EXHIBIT 3

REMARKS

Applicant cancels without prejudice or disclaimer Claims 1-6 and 19-25. Applicant respectfully reserves the right to further prosecute the subject matter of Claims 1-6 and 19-25 in one or more continuing applications. Applicant has amended the Abstract and Summary (paragraph [0003]) to correspond to the scope of subject matter claimed by remaining Claims 7-18. Applicant respectfully requests that the foregoing amendments be entered.

No fees are believed due with this Preliminary Amendment. However, if Applicant has overlooked the need for any fee, the Commissioner is hereby authorized to charge any fees or credit any overpayments associated with this Preliminary Amendment to the deposit account of IBM Corporation, Deposit Account No. 09-0447.

Respectfully submitted,

By: /James L. Baudino/

James L. Baudino
Reg. No. 43,486

Date: February 18, 2013

IBM CORP. (AUS)
c/o The Law Office of James Baudino, PLLC
600 Six Flags Drive
Suite 400
Arlington, TX 76011
Telephone: (817) 751-7140
Fax: (817) 751-7141

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/671,422	11/07/2012	Maria Garza	AUS920120309US1	9343

77351 7590 03/29/2013

IBM CORP. (AUS)
C/O THE LAW OFFICE OF JAMES BAUDINO, PLLC
600 SIX FLAGS DRIVE
SUITE 400
ARLINGTON, TX 76011

EXAMINER	
IPAKCHI, MARYAM M	
ART UNIT	**PAPER NUMBER**
2171	

MAIL DATE	DELIVERY MODE
03/29/2013	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	13/671,422	GARZA ET AL.	
	Examiner	Art Unit	
	MARYAM IPAKCHI	2171	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 November 2012*.

2a)☐ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *7-18* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☐ Claim(s) _____ is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☒ The drawing(s) filed on *November 7, 2012* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date *November 7, 2012*.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

Claims 7-18 are pending, of which claims 7 and 13 are independent. Claims 1-6 and 19-25 are cancelled.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on November 7, 2012 has been considered by the Examiner.

Double Patenting

Claims 7-18 of the instant application are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being anticipated by claims 1-13 of co-pending US Application No. 13/769,593.

The respective conflicting claims while not identical, are not patentably distinct from each other. Claims 7-18 the instant application correspond to products and claims 1-13 of copending US Application No. 13/769,593 correspond to obvious methods of using the products.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the “right to exclude” granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory

obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

US Patent Application No. 13/769,593 to Garza et al.	Instant Application
1.A method, comprising: receiving a request to migrate a running application from a first machine to a second machine; displaying an adjustable resource allocation	7. A system, comprising: a first machine having a running application; a management console operable to manage migrating the running application from the first machine to a second machine, the

mobility setting interface indicating a plurality of mobility settings comprising at least one performance-based mobility setting and at least one concurrency-based mobility setting; receiving, via the interface, a selection of a mobility setting defining a resource allocation to utilize for the migration; and migrating the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting.	management console further operable to: display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration; receive, via the interface, a selection of a mobility setting to apply for migrating the running application; and initiate migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting.
2. The method of Claim 1, further comprising negotiating a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.	8. The system of Claim 7, further comprising at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.
3. The method of Claim 1, further comprising identifying a memory resource allocation for the migration based on the selected mobility setting.	10. The system of Claim 7, further comprising an allocation module operable to identify a memory resource allocation for the migration based on the selected mobility setting.
4. The method of Claim 3, further comprising determining processor utilization for the migration based on the selected mobility setting.	11. The system of Claim 10, wherein the allocation module is operable to identify a processor resource allocation for the migration based on the selected mobility setting.
5. The method of Claim 1, further comprising: allocating greater memory resources for the performance-based mobility setting than for the concurrency-based mobility setting; and allocating a greater quantity of threads for the performance-based mobility setting than for the concurrency-based mobility setting for	12. The system of Claim 7, wherein the interface is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.

managing the memory resources.	
6. The method of Claim 1, further comprising automatically overriding the mobility setting in response to identifying unavailable resources corresponding to a resource allocation indicated by the mobility setting.	9. The system of Claim 8, wherein the VIOS partition is operable to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines.
7. A method, comprising: receiving a request to migrate a plurality of logical partitions from a first machine to a second machine; displaying an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration; receiving, via the interface, a first mobility setting to apply to a first set of logical partitions of the plurality of logical partitions and a second mobility setting to apply to a second set of logical partitions of the plurality of logical partitions; and initiating migration of the first and second sets of logical partitions from the first machine to the second machine utilizing the resource allocations as set by the respective first and second mobility settings.	13. A computer program product for migration operation resource allocation, the computer program product comprising: a computer readable storage medium having computer readable program code embodied therewith, the computer readable program code comprising computer readable program code configured to: receive a request to migrate a running application from a first machine to a second machine; display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration; receive a selection of at least one mobility setting; and migrate the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting.
8. The method of Claim 7, wherein the first	18. The computer program product of Claim 13, wherein the computer

mobility setting is a performance-based mobility setting and the second mobility setting is a concurrency-based mobility setting. 13. The method of Claim 8, further comprising: allocating greater memory resources for the first mobility setting than for the second mobility setting; and allocating a greater quantity of threads for the first mobility setting than for the second mobility setting for managing the memory resources.	readable program code is configured to display the interface to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.
9. The method of Claim 7, further comprising negotiating a balance of resource allocations between the first machine and the second machine based on the first and second mobility settings.	14. The computer program product of Claim 13, wherein the computer readable program code is configured to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.
10. The method of Claim 9, further comprising automatically overriding the resource allocation indicated by either the first or second mobility settings in response to identifying unavailable resources on either the first or second machines. migration based on the first and second mobility settings.	15. The computer program product of Claim 14, wherein the computer readable program code is configured to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines.
11. The method of Claim 7, further comprising identifying a memory resource allocation for the	16. The computer program product of Claim 13, wherein the computer readable program code is configured to identify a memory resource allocation for the migration

	based on the selected mobility setting.
12. The method of Claim 11, further comprising identifying a processor resource allocation for the migration based on the first and second mobility settings.	17. The computer program product of Claim 16, wherein the computer readable program code is configured to identify a processor resource allocation for the migration based on the selected mobility setting.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 7-18 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 6,538,669 to Lagueux, Jr. et al.

Regarding claim 7, Lagueux, Jr. et al. teach:

A system (Lagueux, Jr. et al., col. 1, lines 21-25, col. 3, line 61 – col. 4,line 8)**, comprising:**

a first machine having a running application (Lagueux, Jr. et al., col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … col. 7, lines 6-12 … block storage

interface 118 provides software modules to support block data transfers. Interface 118 includes support for striped data storage, mirrored data storage, partitioned data storage, memory cache storage and RAID storage; col. 9, lines 5-14 …mirroring drivers support intelligent read and write functionality that allow the *storage 150 to be shifted to the new drive array without disrupting access to the data* … *as data is moved to the new storage array, read and write requests for those portions of the data will be directed to the new array while requests for other data portions will be directed to the old storage 150*. Once the migration is complete, the virtual circuit can be adjusted to remove the storage 150 from the virtual circuit.);

a management console operable to manage migrating the running application from the first machine to a second machine, the management console further operable to display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Lagueux, Jr. et al., FIGS. 2-4 and 18-26; col. 2., lines 28-45, col. 3, line 61 - col. 4, line 7 … the device manages the use of a cache memory *to enhance performance;* col. 6, lines 21- 57 … the network interface 146 has network interface specific set of software modules to support configuration, diagnostics, *performance monitoring* …; FIG. 13, col. 16, line 8 - col. 18, line 47 and Table 1, Table 2 … Persistent table module 1400 … can be changed to suit the particular implementation and for certain classes of devices … storage roll call table 1411 includes list of all active storage devices detected … *export table 1407 may include other columns such as current state of the virtual circuit, the capacity of the virtual circuit, etc*. … Table 1 also

shows that a single LUN for a protocol can be connected to different devices depending on the initiator of the storage transaction … ; FIG. 3, col. 10, lines 20-28 … front panel display (FPD) 220 … the FPD provides a user interface for the ISAN server 102A … contains display device and input device … FPD 200 is coupled to HBC modules 202A-B to support status displays, configuration display and management and other management functions, col. 7, lines 13-32 … protocol interface 122 … management interface 120 … contains interfaces for managing access to the tables 116; col. 24, line 8 – col. 25, line 62 …. *Tree of all storage (used and unused) are shown on the display* with each storage element having an icon representing what type it is and some identifying name or ID …storage elements can also be partitioned …);

receive, via the interface, a selection of a mobility setting to apply for migrating the running application (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the ISAN server 102A; col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server I02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example (FIG. 13, col. 16, line 8 - col. 18, line 47) ... The connection option such as the *network interface* 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …);

and

initiate migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting (Lagueux, Jr. et al., col. 9, lines 5-14; col. 22, lines 1-31 ... *storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online* …col. 7, lines 13-20 ... *protocol interface* 122 provides software modules for translating and responding to requests in a variety of protocols ... col. 6, line 17 – col. 9, line 20 … the management interface 120 provides software modules for managing the ISAN server 102A … contains interfaces for managing access to the tables 116 … contains interfaces for rules based management of the system including scheduling or process orchestration, monitoring, informed consent and handling system processes and events).

Regarding claim 8, Lagueux, Jr. et al. teach:

at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting. (*Partition* ….. Lagueux, Jr. et al., FIGS. 2-4, 10, 12, and 18-26, col. 3, line 8-11 … storage elements available for configuration include a logical unit as head of the tree, and a plurality of logical partitions …. col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the ISAN server 102A; col. 14, lines 44-57 … *FIG. 10 illustrates a partition ISM 750* … col.

15, lines 7-50 … FIG. 12 … col. 15, lines 45-50 … *partition ISM modules 1020, 1021* col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server l02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example ... The connection option such as the network interface 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …; col. 24, lines 35045, col. 25, lines 46-54 …. *Storage elements can also be partitioned*) … (*Balance of resource allocations* …. Lagueux, Jr. et al., FIGS. 2-4 and 18-26, … FIG. 16, col. 21, line 38 – col. 22, line 64 … server 1250 includes storage director logic and cache memory …. *Incompatibilities among the plural storage devices and servers can be masked or mimicked as needed using the virtual device architecture.* The *storage director logic utilizing the virtual device architecture provides a single intelligent coordination point for the configuration of server access to storage*. The configuration of the storage server provides accurate configuration information and control by allowing automatic maintenance of the mapping of data ... The storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online. The storage director logic in the storage server operates to consolidate caching requirements from both servers and storage to *reduce the total amount of cache memory required for a storage area network ... system is able to allocate more cache to either the server, the server or storage system than either can effectively provide* as internal memory; FIG. 17 … Storage servers 1300, 1301, 1302 … communication channels 350, 351 … client

servers 1310-1318 communicate with the storage server using storage channel protocols ... according to these protocols, storage transactions are requested, and carry an identifier of the initiator of the request, a LUN, an identifier of the target storage device … resources to emulate the target storage devices so that the client servers smoothly interoperate with the plurality of storage devices in the storage area network).

Regarding claim 9, Lagueux, Jr. et al. teach:

the VIOS partition is operable to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines. (*Partition* ….. Lagueux, Jr. et al., FIGS. 2-4, 10, 12, and 18-26, col. 3, line 8-11 … storage elements available for configuration include a logical unit as head of the tree, and a plurality of logical partitions …. col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the ISAN server 102A; col. 14, lines 44-57 … *FIG. 10 illustrates a partition ISM 750* … col. 15, lines 7-50 … FIG. 12 … col. 15, lines 45-50 … partition ISM modules 1020, 1021 col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server I02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example ... The connection option such as the network interface 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate

virtual device for handling that protocol …; col. 24, lines 35045, col. 25, lines 46-54 …. *Storage elements can also be partitioned*) … (*Override* …. Lagueux, Jr. et al., FIG. 13, col. 16, line 8 - col. 18, line 47 and Table 1, Table 2 … Persistent table module 1400 … can be changed to suit the particular implementation and for certain classes of devices … storage roll call table 1411 includes list of all active storage devices detected … export table 1407 may include other columns such as current state of the virtual circuit, the capacity of the virtual circuit, etc. … Table 1 ... status column indicates status of software or hardware modules ... *status may be "alternate" (i.e., override) if primary device has failed or is not proper … col. 8, lines 52-58 ... ISAN server 102B can be added to the SAN between the server 100A and the storage 150 to provide new functionality and easier administration by supporting storage routing (i.e., override --- new functionality).*

Regarding claim 10, Lagueux, Jr. et al. teach:

an allocation module operable to identify a memory resource allocation for the migration based on the selected mobility setting (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, … FIG. 16, col. 21, line 38 – col. 22, line 64 … server 1250 includes storage director logic and cache memory …. *Incompatibilities among the plural storage devices and servers can be masked or mimicked as needed using the virtual device architecture.* The *storage director logic utilizing the virtual device architecture provides a single intelligent coordination point for the configuration of server access to storage*. The configuration of the storage server provides accurate configuration information and

control by allowing automatic maintenance of the mapping of data ... The storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online. The storage director logic in the storage server operates to *consolidate caching requirements from both servers and storage to reduce the total amount of cache memory required for a storage area network ... system is able to allocate more cache to either the server, the server or storage system than either can effectively provide as internal memory*; further, cache can be dynamically or statically allocated as defined for the applications … FIG. 17 … Storage servers 1300, 1301, 1302 … communication channels 350, 351 … client servers 1310-1318 *communicate with the storage server using storage channel protocols ... according to these protocols, storage transactions are requested*, and carry an identifier of the initiator of the request, a LUN, an identifier of the target storage device … resources to emulate the target storage devices so that the client servers smoothly interoperate with the plurality of storage devices in the storage area network).

Regarding claim 13, Lagueux, Jr. et al. teach:

A computer program product for migration operation resource allocation, the computer program product comprising:

a computer readable storage medium having computer readable program code embodied therewith (Lagueux, Jr. et al., FIG. 4 software module of operating system … col. 11, lines 50-65)**, the computer readable program code comprising computer readable program code configured to:**

receive a request to migrate a running application from a first machine to a second machine (Lagueux, Jr. et al., col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … col. 7, lines 6-12 … block storage interface 118 provides software modules to support block data transfers. Interface 118 includes support for striped data storage, mirrored data storage, partitioned data storage, memory cache storage and RAID storage; col. 9, lines 5-14 …mirroring drivers support intelligent read and write functionality that allow the *storage 150 to be shifted to the new drive array without disrupting access to the data* … as data is moved to the new storage array, read and write requests for those portions of the data will be directed to the new array while requests for other data portions will be directed to the old storage 150. Once the migration is complete, the virtual circuit can be adjusted to remove the storage 150 from the virtual circuit.);

display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the ISAN server 102A; col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server I02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example

(FIG. 13, col. 16, line 8 - col. 18, line 47) ... The connection option such as the network interface 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …);

receive a selection of at least one mobility setting (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the ISAN server 102A; col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server I02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example (FIG. 13, col. 16, line 8 - col. 18, line 47) ... The connection option such as the network interface 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …); **and**

migrate the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting (Lagueux, Jr. et al., col. 9, lines 5-14; col. 22, lines 1-31 ... *storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online* …col. 7, lines 13-20 ... *protocol interface* 122 provides software modules for translating and responding to requests in a variety of protocols ... col. 6, line 17 – col. 9, line 20 … the management interface 120 provides software modules for

managing the ISAN server 102A … contains interfaces for managing access to the tables 116 … contains interfaces for rules based management of the system including scheduling or process orchestration, monitoring, informed consent and handling system processes and events).

Regarding claim 14, Lagueux, Jr. et al. teach:

the computer readable program code is configured to negotiate a balance of resource allocation between the first machine and the second machine based on the selected mobility setting (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, … FIG. 16, col. 21, line 38 – col. 22, line 64 … server 1250 includes storage director logic and cache memory …. *Incompatibilities among the plural storage devices and servers can be masked or mimicked as needed using the virtual device architecture.* The *storage director logic utilizing the virtual device architecture provides a single intelligent coordination point for the configuration of server access to storage*. The configuration of the storage server provides accurate configuration information and control by allowing automatic maintenance of the mapping of data ... The storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online. The storage director logic in the storage server operates to consolidate caching requirements from both servers and storage to *reduce the total amount of cache memory required for a storage area network ... system is able to allocate more cache to either the server, the server or storage system than either can effectively provide* as internal memory; FIG. 17 … Storage servers 1300, 1301, 1302 …

communication channels 350, 351 … client servers 1310-1318 communicate with the storage server using storage channel protocols ... according to these protocols, storage transactions are requested, and carry an identifier of the initiator of the request, a LUN, an identifier of the target storage device … resources to emulate the target storage devices so that the client servers smoothly interoperate with the plurality of storage devices in the storage area network).

Regarding claim 15, Lagueux, Jr. et al. teaches:

the computer readable program code is configured to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines (Lagueux, Jr. et al., FIG. 13, col. 16, line 8 - col. 18, line 47 and Table 1, Table 2 … Persistent table module 1400 … can be changed to suit the particular implementation and for certain classes of devices … storage roll call table 1411 includes list of all active storage devices detected … export table 1407 may include other columns such as current state of the virtual circuit, the capacity of the virtual circuit, etc. … Table 1 ... status column indicates status of software or hardware modules ... *status may be "alternate" (i.e., override) if primary device has failed or is not proper … col. 8, lines 52-58 ... ISAN server 102B can be added to the SAN between the server 100A and the storage 150 to provide new functionality and easier administration by supporting storage routing (i.e., override --- new functionality).*

Regarding claim 16, Lagueux, Jr. et al. teaches:

the computer readable program code is configured to identify a memory resource allocation for the migration based on the selected mobility setting (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, … FIG. 16, col. 21, line 38 – col. 22, line 64 … server 1250 includes storage director logic and cache memory …. *Incompatibilities among the plural storage devices and servers can be masked or mimicked as needed using the virtual device architecture.* The *storage director logic utilizing the virtual device architecture provides a single intelligent coordination point for the configuration of server access to storage*. The configuration of the storage server provides accurate configuration information and control by allowing automatic maintenance of the mapping of data ... The storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online. The storage director logic in the storage server operates to *consolidate caching requirements from both servers and storage to reduce the total amount of cache memory required for a storage area network ... system is able to allocate more cache to either the server, the server or storage system than either can effectively provide as internal memory*; further, cache can be dynamically or statically allocated as defined for the applications … FIG. 17 … Storage servers 1300, 1301, 1302 … communication channels 350, 351 … client servers 1310-1318 *communicate with the storage server using storage channel protocols ... according to these protocols, storage transactions are requested*, and carry an identifier of the initiator of the request, a LUN, an identifier of the target storage device … resources to emulate the target storage devices so that

the client servers smoothly interoperate with the plurality of storage devices in the storage area network).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 11, 12, 17, and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,538,669 to Lagueux, Jr. et al. in view of US Patent Publication No. 2011/0040725 to Sugimoto.

Regarding claim 11, Lagueux, Jr. et al. may fail to explicitly teach each and every feature of:

the allocation module is operable to identify a processor resource allocation for the migration based on the selected mobility setting.

Sugimoto teaches:

the allocation module is operable to identify a processor resource allocation for the migration based on the selected mobility setting (Sugimoto, Abstract … method of increasing a processing performance by setting suitable upper limit of a resources count for each processing request according to an arrangement of

hardware such as a storage device or to contents of the processing request ..., FIGS. 1-3, and 11, database management system 101 … [0037] … resource manager 221 for causing the database management system 101 to hold processes or threads and allocating the processing operation, *an each-request resources-count determiner 217 for determining the number of resources for each processing request*, mapping information 220 of schema and storage arrangement indicating a table definition etc. and the disk device having the data management area which is storing the table definition, etc., and a reference resources count 222 as a threads count which can get the best I/O performance per one disk … [0038] reference resources count 222 … reference-resources-count/disk-device table 224…; [0058]-[0059]).

Lagueux, Jr. et al. relates to a graphical user interface for configuration of a storage system coupled to a storage server. A display and a user input device are included with data processing structures to manage images displayed on the display. The images prompt a use to input configuration data and storage resource data and a data storage transaction received from the communication interface is mapped to one of the configured data paths (Lagueux, Jr. et al., Abstract, col. 1, line 65 - col. 4, line 10). Sugimoto pertains to a database management method for increasing processing performance by setting a suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request (Sugimoto, Abstract).

It would have been obvious to one or ordinary skill in the art at the time of applicants' invention to incorporate the processing performance settings of Sugimoto

with the storage server management architecture of Lagueux, Jr. et al. as a method for improving performances that have been less than desirable and/or optimal. More particularly, Lagueux, Jr. et al. provides for software modules that support performance monitoring and diagnostics (Lagueux, Jr. et al., col. 6, lines 53-58) and, for at least this reason, one of ordinary skill in the art would have considered other approaches for further improving performance.

Regarding claim 12, Lagueux, Jr. et al. may fail to explicitly teach each and every feature of:

the interface is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.

Sugimoto teaches:

the interface is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation (Sugimoto, Abstract … method of increasing a processing performance by setting suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request ..., FIGS. 1-3, and 11, database management system 101 … [0037] … resource manager 221 for causing the database management system 101 to hold processes or threads and allocating the processing operation, *an each-request resources-count determiner 217 for determining*

the number of resources for each processing request, mapping information 220 of schema and storage arrangement indicating a table definition etc. and the disk device having the data management area which is storing the table definition, etc., and a reference resources count 222 as a threads count which can get the best I/O performance per one disk; FIG. 11 (1107) … *[0065]-[0069] … can or cannot be parallelly processed? … when can be parallelly processed, then thread count now used smaller than upper limit? (1102 -1104) before executing (1106) or if cannot be parallelly processed, then execute data operation (1106)…).*

Lagueux, Jr. et al. relates to a graphical user interface for configuration of a storage system coupled to a storage server. A display and a user input device are included with data processing structures to manage images displayed on the display. The images prompt a use to input configuration data and storage resource data and a data storage transaction received from the communication interface is mapped to one of the configured data paths (Lagueux, Jr. et al., Abstract, col. 1, line 65 - col. 4, line 10). Sugimoto pertains to a database management method for increasing processing performance by setting a suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request (Sugimoto, Abstract).

It would have been obvious to one or ordinary skill in the art at the time of applicants' invention to incorporate the processing performance settings of Sugimoto with the storage server management architecture of Lagueux, Jr. et al. as a method for improving performances that have been less than desirable and/or optimal. More

particularly, Lagueux, Jr. et al. provides for software modules that support performance monitoring and diagnostics (Lagueux, Jr. et al., col. 6, lines 53-58) and, for at least this reason, one of ordinary skill in the art would have considered other approaches for further improving performance.

Regarding claim 17, Lagueux, Jr. et al. may fail to explicitly teach each and every feature of:

the computer readable program code is configured to identify a processor resource allocation for the migration based on the selected mobility setting (Sugimoto, Abstract … method of increasing a processing performance by setting suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request ..., FIGS. 1-3, and 11, database management system 101 … [0037] … resource manager 221 for causing the database management system 101 to hold processes or threads and allocating the processing operation, *an each-request resources-count determiner 217 for determining the number of resources for each processing request*, mapping information 220 of schema and storage arrangement indicating a table definition etc. and the disk device having the data management area which is storing the table definition, etc., and a reference resources count 222 as a threads count which can get the best I/O performance per one disk … [0038] reference resources count 222 … reference-resources-count/disk-device table 224…; [0058]-[0059]).

Sugimoto teaches:

the computer readable program code is configured to identify a processor resource allocation for the migration based on the selected mobility setting.

Lagueux, Jr. et al. relates to a graphical user interface for configuration of a storage system coupled to a storage server. A display and a user input device are included with data processing structures to manage images displayed on the display. The images prompt a use to input configuration data and storage resource data and a data storage transaction received from the communication interface is mapped to one of the configured data paths (Lagueux, Jr. et al., Abstract, col. 1, line 65 - col. 4, line 10). Sugimoto pertains to a database management method for increasing processing performance by setting a suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request (Sugimoto, Abstract).

It would have been obvious to one or ordinary skill in the art at the time of applicants' invention to incorporate the processing performance settings of Sugimoto with the storage server management architecture of Lagueux, Jr. et al. as a method for improving performances that have been less than desirable and/or optimal. More particularly, Lagueux, Jr. et al. provides for software modules that support performance monitoring and diagnostics (Lagueux, Jr. et al., col. 6, lines 53-58) and, for at least this reason, one of ordinary skill in the art would have considered other approaches for further improving performance.

Regarding claim 18, Lagueux, Jr. et al. may fail to explicitly teach each and every feature of:

the computer readable program code is configured to display the interface to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.

Sugimoto teaches:

the computer readable program code is configured to display the interface to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation (Sugimoto, Abstract … method of increasing a processing performance by setting suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request ..., FIGS. 1-3, and 11, database management system 101 … [0037] … resource manager 221 for causing the database management system 101 to hold processes or threads and allocating the processing operation, *an each-request resources-count determiner 217 for determining the number of resources for each processing request*, mapping information 220 of schema and storage arrangement indicating a table definition etc. and the disk device having the data management area which is storing the table definition, etc., and a reference resources count 222 as a threads count which can get the best I/O performance per one disk; FIG. 11 (1107) … *[0065]-[0069] … can or cannot be parallelly processed? … when can be*

parallelly processed, then thread count now used smaller than upper limit? (1102 -1104) before executing (1106) or if cannot be parallelly processed, then execute data operation (1106)…).

Lagueux, Jr. et al. relates to a graphical user interface for configuration of a storage system coupled to a storage server. A display and a user input device are included with data processing structures to manage images displayed on the display. The images prompt a use to input configuration data and storage resource data and a data storage transaction received from the communication interface is mapped to one of the configured data paths (Lagueux, Jr. et al., Abstract, col. 1, line 65 - col. 4, line 10). Sugimoto pertains to a database management method for increasing processing performance by setting a suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request (Sugimoto, Abstract).

It would have been obvious to one or ordinary skill in the art at the time of applicants' invention to incorporate the processing performance settings of Sugimoto with the storage server management architecture of Lagueux, Jr. et al. as a method for improving performances that have been less than desirable and/or optimal. More particularly, Lagueux, Jr. et al. provides for software modules that support performance monitoring and diagnostics (Lagueux, Jr. et al., col. 6, lines 53-58) and, for at least this reason, one of ordinary skill in the art would have considered other approaches for further improving performance.

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Examiner has cited particular columns and line numbers (or paragraphs) in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific imitations within the individual claim, other passages and figures of the references may apply as well. More particularly, e.g., in the instances the Examiner has identified Figures of the applied prior art reference, it is understood that the corresponding portion of the written description describing the identified Figures is relied upon. It is respectfully requested from the Applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. The entire reference(s) is/are to be considered to provide disclosure relating to the claimed invention.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MARYAM IPAKCHI whose telephone number is (571)270-3237. The examiner can normally be reached on M-F 9-3:00EST. Any interview requests should be faxed to the examiner at (571)270-4237.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt M. Kim can be reached on (571)272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300. Any interview requests should be faxed to the examiner at (571)270-4237.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MARYAM IPAKCHI
Examiner
Art Unit 2128

/Matt Kim/

Supervisory Patent Examiner, Art Unit 2171

REMARKS

Claims 7-18 were pending at the mailing of an Office Action dated March 29, 2013. Reconsideration and allowance of the claims is respectfully requested.

In the Office Action, the following actions were taken or matters were raised:

Double Patenting

Claims 7-18 stand provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-13 of co-pending Application Serial No. 13/769,593. Given that the above-referenced double patenting rejection is provisional, Applicant respectfully submits that upon the allowance/issuance of either the instant Application and/or the 13/769,593 Application, Applicant will address any non-provisional double patenting rejection maintained by the Examiner.

35 U.S.C. § 102 Rejections

Claims 7-10 and 13-16 stand rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by U.S. Patent No. 6,538,669 of Lagueux et al. (hereinafter "Lagueux"). Applicant respectfully traverses this rejection, and the assertions and determinations therein, for at least the following reasons.

Of the rejected claims, Claims 7 and 13 are independent. Applicant respectfully submits that each of independent Claims 7 and 13 is patentable over the cited reference. For example, independent Claim 7 recites "a first machine having a running application" and "a management console operable to manage **migrating the running application from the first machine to a second machine**." (emphasis added). Applicant respectfully submits that the cited reference does not disclose the above-referenced features of Claim 7.

Lagueux appears to disclose a graphical user interface for the configuration of a storage system. (Lagueux, abstract). Lagueux appears to disclose that data can be moved from one storage array to another storage array. (Lagueux, column 8, line 67 to column 9, line 13). For example, Lagueux recites the following:

> [I]f storage 150 is a terabyte drive array and a new drive array that supports sixteen terabytes of storage is being brought onto the network, the ISAN server 102A can **migrate the data** to the new array without consuming processor time on the server 100A. Further, the mirroring drivers support intelligent read and write functionality that allow the storage 150 to be shifted to the new drive array without disrupting access to the data.
>
> Thus, **as data is moved** to the new storage array, **read and write requests for those portions of the data** will be directed to the new array while requests for other data portions will be directed to the old storage 150.

(Lagueux, column 8, line 67 to column 9, line 13) (emphasis added).

Lagueux does not appear to disclose the migration of a running application. In the Office Action, the Examiner appears to consider the movement of data from one storage array to another storage array as migrating a "running application." (Office action, pages 7 and 8). Applicant respectfully disagrees. The migration of the storage 150 of Lagueux appears to be nothing more than the movement of data identified as storage 150 from one storage array to another storage array. The **movement of data in Lagueux is not the migration of a running application**.

Thus, Applicant respectfully submits that Lagueux does not disclose the above-referenced limitation of Claim 7. Accordingly, for at least this reason, Lagueux does not anticipate Claim 7.

Further, in the Office Action, the Examiner appears to refer to a "persistent table module 1400" as corresponding to the "management console" recited by Claim 7. (Office Action, page 8). Lagueux appears to disclose a persistent table module 1400 that includes a table data access manager 1402. (Lagueux, column 16, lines 713). Lagueux appears to disclose that the table data access manager 1402 and the table class manager 1405 may be used to configure a set of tables (e.g., a fiber channel port ID table, a LUN export table, a configuration template table, etc.). (Lagueux, column 16, lines 7-25). Thus, **the persistent table module 1400 of Lagueux does not appear to be used to manage the migration of a running application. Nor does the persistent table module 1400 of Lagueux appear to display an interface comprising "selectable mobility settings" that correspond to a desired resource allocation to apply for**

migrating a running application. Thus, for this reason also, Lagueux does not appear to disclose the limitations of Claim 7.

Further, the Examiner also appears to rely on a front panel display (FPD) 220 of an ISAN server 102A of Lagueux to perhaps correspond to the "management console" recited by Claim 7. (Office Action, page 9). The FPD appears to include a user interface that supports status displays, configuration display and management, and other management functions. (Lagueux, column 10, lines 21-28). However, **nowhere in Lagueux does there appear to be any teaching that the FPD of Lagueux displays "selectable mobility settings" each corresponding to "a desired resource allocation to utilize" for the migration of a running application**.

Additionally, Lagueux does not appear to disclose the display or the receipt of a selection of a "mobility setting" that corresponds to "a desired resource allocation" to utilize for migrating a running application from one machine to another machine. In the Office action, the Examiner appears to rely on column 18, lines 51-61, as disclosing the above-referenced limitations recited by Claim 7. (Office Action, page 9). Applicant respectfully disagrees.

Lagueux appears to disclose a representative ISAN server 102 (figure 2) that may have a number of connection options (e.g., serial 140, front panel 142, Ethernet 144, and network interface card (NIC) 146). (Lagueux, figure 2, column 6, lines 39-58). The Examiner appears to rely on **an example storage transaction** that, in this particular example, is coming over the network interface 146. (Office action, page 9 (referring to Lagueux, column 18, lines 51-61)). In this particular example, Lagueux appears to disclose that, for a storage transaction arriving over the network interface, hardware device driver receives the transaction and dispatches it to an appropriate virtual device based on a protocol. (Lagueux, column 18, lines 51-61). Neither in the portion of Lagueux referenced by the Examiner, nor elsewhere in Lagueux, does there appear to be any disclosure of the display or the receipt of a selection of a "mobility setting" that corresponds to "a desired resource allocation" to utilize for migrating a running application from one machine to another machine as recited by Claim 7. Therefore, for at least this reason also, Lagueux does not appear to anticipate Claim 7.

Accordingly, for at least these reasons, Lagueux does not appear to disclose each and every limitation recited by Claim 7. Therefore, Applicant respectfully requests that the rejection of Claim 7 be withdrawn.

Independent Claim 13 recites computer readable program code configured to "receive a request to **migrate a running application from a first machine to a second machine**," "display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, **each mobility setting corresponding to a desired resource allocation to utilize for the migration**," "receive a selection of at least one mobility setting," and "**migrate the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting**." (emphasis added). At least for the reasons indicated above in connection with independent Claim 7, Applicant respectfully submits that the cited reference does not disclose each and every limitation recited by Claim 13. Accordingly, Applicant respectfully requests that the rejection of Claim 13 be withdrawn.

Claims 8-10 and 14-16 that depend respectively from independent Claims 7 and 13 are also not anticipated by Lagueux at least because they incorporate the limitations of respective Claims 7 and 13 and also add additional elements that further distinguish Lagueux. Therefore, Applicant respectfully requests that the rejection of Claims 7-10 and 13-16 be withdrawn.

35 U.S.C. § 103 Rejections

Claims 11, 12, 17 and 18 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Lagueux in view of U.S. Patent Publication No. 2011/0040725 of Sugimoto (hereinafter "Sugimoto"). Applicant respectfully traverses this rejection and the assertions and determinations therein, for at least the following reasons.

Claims 11, 12, 17 and 18 depend from respective independent Claims 7 and 13. As indicated above, Applicant respectfully submits that independent Claims 7 and 13 are patentable over the cited Lagueux reference. Therefore, Claims 11, 12, 17 and 18 that depend from respective independent Claims 7 and 13 are also patentable. Further, Sugimoto does not appear to remedy the deficiencies of Lagueux indicated above. Therefore, Applicant respectfully requests that the rejection of Claims 11, 12, 17 and 18 be withdrawn.

CONCLUSION

Having now responded to each rejection set forth in the present Office Action, Applicant respectfully submits that all claims now pending are in condition for allowance and respectfully requests such allowance.

No fee is believed due with this Response. If Applicant has overlooked the need for any fee due with this Response, the Commissioner is hereby authorized to charge any fees or credit any overpayments associated with this Response to the deposit account of IBM Corporation, Deposit Account No. 09-0447.

Respectfully submitted,

By: /James L. Baudino/

James L. Baudino
Reg. No. 43,486

Date: July 1, 2013

IBM Corporation (AUS)
c/o The Law Office of James Baudino, PLLC
600 Six Flags Drive
Suite 400
Arlington, Texas 76011
Phone: 817-751-7140
Fax: 817-751-7141

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/671,422	11/07/2012	Maria Garza	AUS920120309US1	9343

77351 7590 08/02/2013

IBM CORP. (AUS)
C/O THE LAW OFFICE OF JAMES BAUDINO, PLLC
600 SIX FLAGS DRIVE
SUITE 400
ARLINGTON, TX 76011

EXAMINER	
IPAKCHI, MARYAM M	
ART UNIT	**PAPER NUMBER**
2171	
MAIL DATE	**DELIVERY MODE**
08/02/2013	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 13/671,422	Applicant(s) GARZA ET AL.	
	Examiner MARYAM IPAKCHI	Art Unit 2171	AIA (First Inventor to File) Status No

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *1 July 2013*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

5)☒ Claim(s) *7-18* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *7-18* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some * c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

Claims 7-18 are pending, of which claims 7 and 13 are independent. Claims 1-6 and 19-25 are cancelled.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on November 7, 2012 has been considered by the Examiner.

Response to Arguments

Applicant's arguments filed on July 1, 2013 have been fully considered but they are *not* persuasive for at least the reasons set forth below.

The provisional obviousness-type double patenting rejection of claims 7-18 over claims 1-13 is maintained as no amendments/cancellations were made to any of the relevant claims and/or no terminal disclaimer was filed.

The 35 USC 102 and 103 rejections of claims 7-18 are maintained.

Applicants' assert, at pages 9-10 of Applicants' July 1, 2013, filed response that Lagueux, Jr. et al. allegedly fails to disclose or suggest:

> "a management console operable to manage migrating the running application" and to initiate "migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting" as recited in independent claim 7, and as similarly recited in independent claim 13.

Examiner respectfully disagrees. It is respectfully noted that during examination, a claim **must be given its broadest reasonable interpretation consistent with the specification**. Under a broadest reasonable interpretation, *words of the claim must be given their plain meaning,* unless such meaning is *inconsistent* with the specification. M.P.E.P. 2173.01(I). *Under broadest reasonable interpretation approach*, "migration of a running application" may, reasonably and broadly, be understood as migrating data associated, at any level, with a running application. That is, an application is inherently processing/manipulating data and the claim does not provide any metes or bounds regarding what extent and/or feature(s) of the application is being migrated. More particularly, e.g., does migration of data correspond to the entire application itself, data corresponding to part of the application itself, information being processed by the application, a combination of information being processed and some or all of the application itself. It is noted that such an interpretation is consistent with Applicants' specification as the specification utilizes the term 'program instructions' themselves, as opposed to "application." **The claim language as it now stands is open-ended and not specific regarding the metes and bounds of the claim term "application."**

Further, Lagueux discloses storage management applications such that the application of Lagueux is, if at all possible, even more rooted with the data in storage (Lagueux et al., col. 4, lines 5-8).

Additionally, with regard to Applicants' comments, at pages 6-7 of Applicants' Response, regarding the "management console," it is respectfully submitted Lagueux et al. does not merely teach a table module 1400, but also discloses interface (118-122),

including a block storage interface 118, network interface 146, **a management interface** 120 and a protocol interface 122 (Lagueux et al., FIG. 18, col. 6, lines 21-31). The **management interface 120 provides rules for managing the server 102A and access to tables 116 and interfaces for management of system including process orchestration, monitoring, handling processes and events** (col. 7, lines 21-31). Lagueux et al. also provides connection options 130 and hardware interface 126 (col. 6, lines 54-64) and even discusses , one or more of which may be It is respectfully submitted that claim 7 recites merely provides 'a mobility setting corresponding to a desired resource allocation to utilize for the migration,' and the claim language itself may be reasonably interpreted such as selecting one of a plurality of storage resources coupled to one of a plurality of data paths and/or communication interfaces (Lagueux et al., Abstract).

For at least the reasons set forth herein, it is respectfully submitted that Lagueux et al. teaches each and every feature of independent claims 7 and 13.

Double Patenting

Claims 7-18 of the instant application are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being anticipated by claims 1-13 of co-pending US Application No. 13/769,593.

The respective conflicting claims while not identical, are not patentably distinct from each other. Claims 7-18 the instant application correspond to products and claims

1-13 of copending US Application No. 13/769,593 correspond to obvious methods of using the products.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

US Patent Application No. 13/769,593 to Garza et al.	Instant Application
1.A method, comprising: receiving a request to migrate a running application from a first machine to a second machine; displaying an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings comprising at least one performance-based mobility setting and at least one concurrency-based mobility setting; receiving, via the interface, a selection of a mobility setting defining a resource allocation to utilize for the migration; and migrating the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting.	7. A system, comprising: a first machine having a running application; a management console operable to manage migrating the running application from the first machine to a second machine, the management console further operable to: display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration; receive, via the interface, a selection of a mobility setting to apply for migrating the running application; and initiate migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting.
2. The method of Claim 1, further comprising negotiating a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.	8. The system of Claim 7, further comprising at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.
3. The method of Claim 1, further comprising identifying a memory resource allocation for the migration based on the selected mobility setting.	10. The system of Claim 7, further comprising an allocation module operable to identify a memory resource allocation for the migration based on the selected mobility setting.

4. The method of Claim 3, further comprising determining processor utilization for the migration based on the selected mobility setting.	11. The system of Claim 10, wherein the allocation module is operable to identify a processor resource allocation for the migration based on the selected mobility setting.
5. The method of Claim 1, further comprising: allocating greater memory resources for the performance-based mobility setting than for the concurrency-based mobility setting; and allocating a greater quantity of threads for the performance-based mobility setting than for the concurrency-based mobility setting for managing the memory resources.	12. The system of Claim 7, wherein the interface is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.
6. The method of Claim 1, further comprising automatically overriding the mobility setting in response to identifying unavailable resources corresponding to a resource allocation indicated by the mobility setting.	9. The system of Claim 8, wherein the VIOS partition is operable to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines.
7. A method, comprising: receiving a request to migrate a plurality of logical partitions from a first machine to a second machine; displaying an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration; receiving, via the interface, a first mobility setting to apply to a first set of logical partitions of the plurality of logical partitions and a second mobility setting to apply to a second set	13. A computer program product for migration operation resource allocation, the computer program product comprising: a computer readable storage medium having computer readable program code embodied therewith, the computer readable program code comprising computer readable program code configured to: receive a request to migrate a running application from a first machine to a second machine; display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource

of logical partitions of the plurality of logical partitions; and initiating migration of the first and second sets of logical partitions from the first machine to the second machine utilizing the resource allocations as set by the respective first and second mobility settings.	allocation to utilize for the migration; receive a selection of at least one mobility setting; and migrate the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting.
8. The method of Claim 7, wherein the first mobility setting is a performance-based mobility setting and the second mobility setting is a concurrency-based mobility setting. 13. The method of Claim 8, further comprising: allocating greater memory resources for the first mobility setting than for the second mobility setting; and allocating a greater quantity of threads for the first mobility setting than for the second mobility setting for managing the memory resources.	18. The computer program product of Claim 13, wherein the computer readable program code is configured to display the interface to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.
9. The method of Claim 7, further comprising negotiating a balance of resource allocations between the first machine and the second machine based on the first and second mobility settings.	14. The computer program product of Claim 13, wherein the computer readable program code is configured to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.
10. The method of Claim 9, further comprising automatically overriding the resource allocation	15. The computer program product of Claim 14, wherein the computer

indicated by either the first or second mobility settings in response to identifying unavailable resources on either the first or second machines. migration based on the first and second mobility settings.	readable program code is configured to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines.
11. The method of Claim 7, further comprising identifying a memory resource allocation for the	16. The computer program product of Claim 13, wherein the computer readable program code is configured to identify a memory resource allocation for the migration based on the selected mobility setting.
12. The method of Claim 11, further comprising identifying a processor resource allocation for the migration based on the first and second mobility settings.	17. The computer program product of Claim 16, wherein the computer readable program code is configured to identify a processor resource allocation for the migration based on the selected mobility setting.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 7-18 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 6,538,669 to Lagueux, Jr. et al.

Regarding claim 7, Lagueux, Jr. et al. teach:

A system (Lagueux, Jr. et al., col. 1, lines 21-25, col. 3, line 61 – col. 4,line 8)**, comprising:**

a first machine having a running application (Lagueux, Jr. et al., col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … col. 7, lines 6-12 … block storage interface 118 provides software modules to support block data transfers. Interface 118 includes support for striped data storage, mirrored data storage, partitioned data storage, memory cache storage and RAID storage; col. 9, lines 5-14 …mirroring drivers support intelligent read and write functionality that allow the *storage 150 to be shifted to the new drive array without disrupting access to the data … as data is moved to the new storage array, read and write requests for those portions of the data will be directed to the new array while requests for other data portions will be directed to the old storage 150*. Once the migration is complete, the virtual circuit can be adjusted to remove the storage 150 from the virtual circuit.);

a management console operable to manage migrating the running application from the first machine to a second machine, the management console further operable to display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Lagueux, Jr. et al., FIGS. 2-4 and 18-26; col. 2., lines 28-45, col. 3, line 61 - col. 4, line 7 … the device manages the use of a cache

memory *to enhance performance;* col. 6, lines 21- 57 … the network interface 146 has network interface specific set of software modules to support configuration, diagnostics, *performance monitoring* …; FIG. 13, col. 16, line 8 - col. 18, line 47 and Table 1, Table 2 … Persistent table module 1400 … can be changed to suit the particular implementation and for certain classes of devices … storage roll call table 1411 includes list of all active storage devices detected … *export table 1407 may include other columns such as current state of the virtual circuit, the capacity of the virtual circuit, etc*. … Table 1 also shows that a single LUN for a protocol can be connected to different devices depending on the initiator of the storage transaction … ; FIG. 3, col. 10, lines 20-28 … front panel display (FPD) 220 … the FPD provides a user interface for the ISAN server 102A … contains display device and input device … FPD 200 is coupled to HBC modules 202A-B to support status displays, configuration display and management and other management functions, col. 7, lines 13-32 … protocol interface 122 … management interface 120 … contains interfaces for managing access to the tables 116; col. 24, line 8 – col. 25, line 62 …. *Tree of all storage (used and unused) are shown on the display* with each storage element having an icon representing what type it is and some identifying name or ID …storage elements can also be partitioned …);

receive, via the interface, a selection of a mobility setting to apply for migrating the running application (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the

ISAN server 102A; col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server I02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example (FIG. 13, col. 16, line 8 - col. 18, line 47) ... The connection option such as the *network interface* 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …);

and

initiate migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting

(Lagueux, Jr. et al., col. 9, lines 5-14; col. 22, lines 1-31 ... *storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online* …col. 7, lines 13-20 ... *protocol interface* 122 provides software modules for translating and responding to requests in a variety of protocols ... col. 6, line 17 – col. 9, line 20 … the management interface 120 provides software modules for managing the ISAN server 102A … contains interfaces for managing access to the tables 116 … contains interfaces for rules based management of the system including scheduling or process orchestration, monitoring, informed consent and handling system processes and events).

Regarding claim 8, Lagueux, Jr. et al. teach:

at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the

second machine based on the selected mobility setting. (*Partition* ….. Lagueux, Jr. et al., FIGS. 2-4, 10, 12, and 18-26, col. 3, line 8-11 … storage elements available for configuration include a logical unit as head of the tree, and a plurality of logical partitions …. col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the ISAN server 102A; col. 14, lines 44-57 … *FIG. 10 illustrates a partition ISM 750* … col. 15, lines 7-50 … FIG. 12 … col. 15, lines 45-50 … *partition ISM modules 1020, 1021* col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server l02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example ... The connection option such as the network interface 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …; col. 24, lines 35045, col. 25, lines 46-54 …. *Storage elements can also be partitioned*) … (*Balance of resource allocations* …. Lagueux, Jr. et al., FIGS. 2-4 and 18-26, … FIG. 16, col. 21, line 38 – col. 22, line 64 … server 1250 includes storage director logic and cache memory …. *Incompatibilities among the plural storage devices and servers can be masked or mimicked as needed using the virtual device architecture.* The *storage director logic utilizing the virtual device architecture provides a single intelligent coordination point for the configuration of server access to storage*. The configuration of the storage server provides accurate configuration information and control by allowing automatic maintenance of the mapping of data ...

The storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online. The storage director logic in the storage server operates to consolidate caching requirements from both servers and storage to *reduce the total amount of cache memory required for a storage area network ... system is able to allocate more cache to either the server, the server or storage system than either can effectively provide* as internal memory; FIG. 17 … Storage servers 1300, 1301, 1302 … communication channels 350, 351 … client servers 1310-1318 communicate with the storage server using storage channel protocols ... according to these protocols, storage transactions are requested, and carry an identifier of the initiator of the request, a LUN, an identifier of the target storage device … resources to emulate the target storage devices so that the client servers smoothly interoperate with the plurality of storage devices in the storage area network).

Regarding claim 9, Lagueux, Jr. et al. teach:

the VIOS partition is operable to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines. (*Partition* ….. Lagueux, Jr. et al., FIGS. 2-4, 10, 12, and 18-26, col. 3, line 8-11 … storage elements available for configuration include a logical unit as head of the tree, and a plurality of logical partitions …. col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage

routing functionality of the ISAN server 102A; col. 14, lines 44-57 … *FIG. 10 illustrates a partition ISM 750* … col. 15, lines 7-50 … FIG. 12 … col. 15, lines 45-50 … partition ISM modules 1020, 1021 col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server I02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example ... The connection option such as the network interface 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …; col. 24, lines 35045, col. 25, lines 46-54 …. *Storage elements can also be partitioned*) … (*Override* …. Lagueux, Jr. et al., FIG. 13, col. 16, line 8 - col. 18, line 47 and Table 1, Table 2 … Persistent table module 1400 … can be changed to suit the particular implementation and for certain classes of devices … storage roll call table 1411 includes list of all active storage devices detected … export table 1407 may include other columns such as current state of the virtual circuit, the capacity of the virtual circuit, etc. … Table 1 ... status column indicates status of software or hardware modules ... *status may be "alternate" (i.e., override) if primary device has failed or is not proper … col. 8, lines 52-58 ... ISAN server 102B can be added to the SAN between the server 100A and the storage 150 to provide new functionality and easier administration by supporting storage routing (i.e., override --- new functionality).*

Regarding claim 10, Lagueux, Jr. et al. teach:

an allocation module operable to identify a memory resource allocation for the migration based on the selected mobility setting (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, … FIG. 16, col. 21, line 38 – col. 22, line 64 … server 1250 includes storage director logic and cache memory …. *Incompatibilities among the plural storage devices and servers can be masked or mimicked as needed using the virtual device architecture.* The *storage director logic utilizing the virtual device architecture provides a single intelligent coordination point for the configuration of server access to storage*. The configuration of the storage server provides accurate configuration information and control by allowing automatic maintenance of the mapping of data ... The storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online. The storage director logic in the storage server operates to *consolidate caching requirements from both servers and storage to reduce the total amount of cache memory required for a storage area network ... system is able to allocate more cache to either the server, the server or storage system than either can effectively provide as internal memory*; further, cache can be dynamically or statically allocated as defined for the applications … FIG. 17 … Storage servers 1300, 1301, 1302 … communication channels 350, 351 … client servers 1310-1318 *communicate with the storage server using storage channel protocols ... according to these protocols, storage transactions are requested*, and carry an identifier of the initiator of the request, a LUN, an identifier of the target storage device … resources to emulate the target storage devices so that the client servers smoothly interoperate with the plurality of storage devices in the storage area network).

Regarding claim 13, Lagueux, Jr. et al. teach:

A computer program product for migration operation resource allocation, the computer program product comprising:

a computer readable storage medium having computer readable program code embodied therewith (Lagueux, Jr. et al., FIG. 4 software module of operating system … col. 11, lines 50-65)**, the computer readable program code comprising computer readable program code configured to:**

receive a request to migrate a running application from a first machine to a second machine (Lagueux, Jr. et al., col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … col. 7, lines 6-12 … block storage interface 118 provides software modules to support block data transfers. Interface 118 includes support for striped data storage, mirrored data storage, partitioned data storage, memory cache storage and RAID storage; col. 9, lines 5-14 …mirroring drivers support intelligent read and write functionality that allow the *storage 150 to be shifted to the new drive array without disrupting access to the data* … as data is moved to the new storage array, read and write requests for those portions of the data will be directed to the new array while requests for other data portions will be directed to the old storage 150. Once the migration is complete, the virtual circuit can be adjusted to remove the storage 150 from the virtual circuit.);

display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the ISAN server 102A; col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server I02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example (FIG. 13, col. 16, line 8 - col. 18, line 47) ... The connection option such as the network interface 146 over which the storage transaction is received ... hardware device driver, depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …);

receive a selection of at least one mobility setting (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, col. 7, lines 33-36 …. Storage transactions include read and write requests as well as status inquiries … requests may be block oriented … operating system 124 …. and the interfaces 118-122 support the virtual device and storage routing functionality of the ISAN server 102A; col. 18, lines 51-67 … storage transaction that comes over one of the connection options 130 to the ISAN server I02A ... Assume that the ISAN server 102A is configured as shown in Tables 1 and 2 for this example (FIG. 13, col. 16, line 8 - col. 18, line 47) ... The connection option such as the network interface 146 over which the storage transaction is received ... hardware device driver,

depending on the protocol, dispatches it to an appropriate virtual device for handling that protocol …); **and**

migrate the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting (Lagueux, Jr. et al., col. 9, lines 5-14; col. 22, lines 1-31 ... *storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online* …col. 7, lines 13-20 ... *protocol interface* 122 provides software modules for translating and responding to requests in a variety of protocols ... col. 6, line 17 – col. 9, line 20 … the management interface 120 provides software modules for managing the ISAN server 102A … contains interfaces for managing access to the tables 116 … contains interfaces for rules based management of the system including scheduling or process orchestration, monitoring, informed consent and handling system processes and events).

Regarding claim 14, Lagueux, Jr. et al. teach:

the computer readable program code is configured to negotiate a balance of resource allocation between the first machine and the second machine based on the selected mobility setting (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, … FIG. 16, col. 21, line 38 – col. 22, line 64 … server 1250 includes storage director logic and cache memory …. *Incompatibilities among the plural storage devices and servers can be masked or mimicked as needed using the virtual device architecture.* The *storage director logic utilizing the virtual device architecture provides a single intelligent*

coordination point for the configuration of server access to storage. The configuration of the storage server provides accurate configuration information and control by allowing automatic maintenance of the mapping of data ... The storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online. The storage director logic in the storage server operates to consolidate caching requirements from both servers and storage to *reduce the total amount of cache memory required for a storage area network ... system is able to allocate more cache to either the server, the server or storage system than either can effectively provide* as internal memory; FIG. 17 … Storage servers 1300, 1301, 1302 … communication channels 350, 351 … client servers 1310-1318 communicate with the storage server using storage channel protocols ... according to these protocols, storage transactions are requested, and carry an identifier of the initiator of the request, a LUN, an identifier of the target storage device … resources to emulate the target storage devices so that the client servers smoothly interoperate with the plurality of storage devices in the storage area network).

Regarding claim 15, Lagueux, Jr. et al. teach:

the computer readable program code is configured to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines (Lagueux, Jr. et al., FIG. 13, col. 16, line 8 - col. 18, line 47 and Table 1, Table 2 … Persistent table module 1400 … can be changed to suit the

particular implementation and for certain classes of devices … storage roll call table 1411 includes list of all active storage devices detected … export table 1407 may include other columns such as current state of the virtual circuit, the capacity of the virtual circuit, etc. … Table 1 ... status column indicates status of software or hardware modules ... *status may be "alternate" (i.e., override) if primary device has failed or is not proper … col. 8, lines 52-58 ... ISAN server 102B can be added to the SAN between the server 100A and the storage 150 to provide new functionality and easier administration by supporting storage routing (i.e., override --- new functionality).*

Regarding claim 16, Lagueux, Jr. et al. teach:

the computer readable program code is configured to identify a memory resource allocation for the migration based on the selected mobility setting (Lagueux, Jr. et al., FIGS. 2-4 and 18-26, … FIG. 16, col. 21, line 38 – col. 22, line 64 … server 1250 includes storage director logic and cache memory …. *Incompatibilities among the plural storage devices and servers can be masked or mimicked as needed using the virtual device architecture.* The *storage director logic utilizing the virtual device architecture provides a single intelligent coordination point for the configuration of server access to storage*. The configuration of the storage server provides accurate configuration information and control by allowing automatic maintenance of the mapping of data ... The storage director at the server provides for active migration of data from old storage devices to new storage devices while the devices remain online. The storage director logic in the storage server operates to *consolidate caching*

requirements from both servers and storage to reduce the total amount of cache memory required for a storage area network ... system is able to allocate more cache to either the server, the server or storage system than either can effectively provide as internal memory; further, cache can be dynamically or statically allocated as defined for the applications … FIG. 17 … Storage servers 1300, 1301, 1302 … communication channels 350, 351 … client servers 1310-1318 *communicate with the storage server using storage channel protocols ... according to these protocols, storage transactions are requested*, and carry an identifier of the initiator of the request, a LUN, an identifier of the target storage device … resources to emulate the target storage devices so that the client servers smoothly interoperate with the plurality of storage devices in the storage area network).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 11, 12, 17, and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,538,669 to Lagueux, Jr. et al. in view of US Patent Publication No. 2011/0040725 to Sugimoto.

Regarding claim 11, Lagueux, Jr. et al. may fail to explicitly teach each and every feature of:

the allocation module is operable to identify a processor resource allocation for the migration based on the selected mobility setting.

Sugimoto teaches:

the allocation module is operable to identify a processor resource allocation for the migration based on the selected mobility setting (Sugimoto, Abstract … method of increasing a processing performance by setting suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request ..., FIGS. 1-3, and 11, database management system 101 … [0037] … resource manager 221 for causing the database management system 101 to hold processes or threads and allocating the processing operation, *an each-request resources-count determiner 217 for determining the number of resources for each processing request*, mapping information 220 of schema and storage arrangement indicating a table definition etc. and the disk device having the data management area which is storing the table definition, etc., and a reference resources count 222 as a threads count which can get the best I/O performance per one disk … [0038] reference resources count 222 … reference-resources-count/disk-device table 224…; [0058]-[0059]).

Lagueux, Jr. et al. relates to a graphical user interface for configuration of a storage system coupled to a storage server. A display and a user input device are included with data processing structures to manage images displayed on the display.

The images prompt a use to input configuration data and storage resource data and a data storage transaction received from the communication interface is mapped to one of the configured data paths (Lagueux, Jr. et al., Abstract, col. 1, line 65 - col. 4, line 10). Sugimoto pertains to a database management method for increasing processing performance by setting a suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request (Sugimoto, Abstract).

It would have been obvious to one or ordinary skill in the art at the time of applicants' invention to incorporate the processing performance settings of Sugimoto with the storage server management architecture of Lagueux, Jr. et al. as a method for improving performances that have been less than desirable and/or optimal. More particularly, Lagueux, Jr. et al. provides for software modules that support performance monitoring and diagnostics (Lagueux, Jr. et al., col. 6, lines 53-58) and, for at least this reason, one of ordinary skill in the art would have considered other approaches for further improving performance.

Regarding claim 12, Lagueux, Jr. et al. may fail to explicitly teach each and every feature of:

the interface is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.

Sugimoto teaches:

the interface is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation (Sugimoto, Abstract … method of increasing a processing performance by setting suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request ..., FIGS. 1-3, and 11, database management system 101 … [0037] … resource manager 221 for causing the database management system 101 to hold processes or threads and allocating the processing operation, *an each-request resources-count determiner 217 for determining the number of resources for each processing request*, mapping information 220 of schema and storage arrangement indicating a table definition etc. and the disk device having the data management area which is storing the table definition, etc., and a reference resources count 222 as a threads count which can get the best I/O performance per one disk; FIG. 11 (1107) … *[0065]-[0069] … can or cannot be parallelly processed? … when can be parallelly processed, then thread count now used smaller than upper limit? (1102 -1104) before executing (1106) or if cannot be parallelly processed, then execute data operation (1106)…).*

Lagueux, Jr. et al. relates to a graphical user interface for configuration of a storage system coupled to a storage server. A display and a user input device are included with data processing structures to manage images displayed on the display. The images prompt a use to input configuration data and storage resource data and a data storage transaction received from the communication interface is mapped to one of

the configured data paths (Lagueux, Jr. et al., Abstract, col. 1, line 65 - col. 4, line 10). Sugimoto pertains to a database management method for increasing processing performance by setting a suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request (Sugimoto, Abstract).

It would have been obvious to one or ordinary skill in the art at the time of applicants' invention to incorporate the processing performance settings of Sugimoto with the storage server management architecture of Lagueux, Jr. et al. as a method for improving performances that have been less than desirable and/or optimal. More particularly, Lagueux, Jr. et al. provides for software modules that support performance monitoring and diagnostics (Lagueux, Jr. et al., col. 6, lines 53-58) and, for at least this reason, one of ordinary skill in the art would have considered other approaches for further improving performance.

Regarding claim 17, Lagueux, Jr. et al. may fail to explicitly teach each and every feature of:

the computer readable program code is configured to identify a processor resource allocation for the migration based on the selected mobility setting (Sugimoto, Abstract … method of increasing a processing performance by setting suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request ..., FIGS. 1-3, and 11, database management system 101 … [0037] … resource manager 221 for

causing the database management system 101 to hold processes or threads and allocating the processing operation, *an each-request resources-count determiner 217 for determining the number of resources for each processing request*, mapping information 220 of schema and storage arrangement indicating a table definition etc. and the disk device having the data management area which is storing the table definition, etc., and a reference resources count 222 as a threads count which can get the best I/O performance per one disk … [0038] reference resources count 222 … reference-resources-count/disk-device table 224…; [0058]-[0059]).

Sugimoto teaches:

the computer readable program code is configured to identify a processor resource allocation for the migration based on the selected mobility setting.

Lagueux, Jr. et al. relates to a graphical user interface for configuration of a storage system coupled to a storage server. A display and a user input device are included with data processing structures to manage images displayed on the display. The images prompt a use to input configuration data and storage resource data and a data storage transaction received from the communication interface is mapped to one of the configured data paths (Lagueux, Jr. et al., Abstract, col. 1, line 65 - col. 4, line 10). Sugimoto pertains to a database management method for increasing processing performance by setting a suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request (Sugimoto, Abstract).

It would have been obvious to one or ordinary skill in the art at the time of applicants' invention to incorporate the processing performance settings of Sugimoto with the storage server management architecture of Lagueux, Jr. et al. as a method for improving performances that have been less than desirable and/or optimal. More particularly, Lagueux, Jr. et al. provides for software modules that support performance monitoring and diagnostics (Lagueux, Jr. et al., col. 6, lines 53-58) and, for at least this reason, one of ordinary skill in the art would have considered other approaches for further improving performance.

Regarding claim 18, Lagueux, Jr. et al. may fail to explicitly teach each and every feature of:

the computer readable program code is configured to display the interface to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.

Sugimoto teaches:

the computer readable program code is configured to display the interface to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation (Sugimoto, Abstract … method of increasing a processing performance by setting suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device

or to contents of the processing request ..., FIGS. 1-3, and 11, database management system 101 … [0037] … resource manager 221 for causing the database management system 101 to hold processes or threads and allocating the processing operation, *an each-request resources-count determiner 217 for determining the number of resources for each processing request*, mapping information 220 of schema and storage arrangement indicating a table definition etc. and the disk device having the data management area which is storing the table definition, etc., and a reference resources count 222 as a threads count which can get the best I/O performance per one disk; FIG. 11 (1107) … *[0065]-[0069] … can or cannot be parallelly processed? … when can be parallelly processed, then thread count now used smaller than upper limit? (1102 -1104) before executing (1106) or if cannot be parallelly processed, then execute data operation (1106)…).*

Lagueux, Jr. et al. relates to a graphical user interface for configuration of a storage system coupled to a storage server. A display and a user input device are included with data processing structures to manage images displayed on the display. The images prompt a use to input configuration data and storage resource data and a data storage transaction received from the communication interface is mapped to one of the configured data paths (Lagueux, Jr. et al., Abstract, col. 1, line 65 - col. 4, line 10). Sugimoto pertains to a database management method for increasing processing performance by setting a suitable upper limit of a resources count for each processing request according to an arrangement of hardware such as a storage device or to contents of the processing request (Sugimoto, Abstract).

It would have been obvious to one or ordinary skill in the art at the time of applicants' invention to incorporate the processing performance settings of Sugimoto with the storage server management architecture of Lagueux, Jr. et al. as a method for improving performances that have been less than desirable and/or optimal. More particularly, Lagueux, Jr. et al. provides for software modules that support performance monitoring and diagnostics (Lagueux, Jr. et al., col. 6, lines 53-58) and, for at least this reason, one of ordinary skill in the art would have considered other approaches for further improving performance.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a). A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Examiner has cited particular columns and line numbers (or paragraphs) in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific imitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the Applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. **The entire reference(s) is/are to be considered to provide disclosure relating to the claimed invention.**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MARYAM IPAKCHI whose telephone number is (571)270-3237. The examiner can normally be reached on M-F 7-3:00EST. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Omar F. Fernandez Rivas can be reached on 571-272-2589. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

MARYAM IPAKCHI
Examiner
Art Unit 2171

/Matt Kim/

Supervisory Patent Examiner, Art Unit 2171

REMARKS

Claims 7-18 and 26-32 were pending at the mailing of an Office Action dated June 12, 2014. Reconsideration and allowance of the claims is respectfully requested.

Applicant thanks Examiner Ipakchi for conducting a telephone interview with Applicant's representative James L. Baudino (Reg. 43,486) on September 8, 2014, to briefly discuss an office action in related application serial no. 13/769,593 and, specifically, the Section 102 and 103 rejections. In the interview, a discussion of the claimed subject matter was discussed in view of the cited references, such as the migration of an operating system instance, application and/or LPAR and the interface for selectable mobility settings. No agreement was reached. However, Applicant uses the information from the above-referenced interview to respond herewith. Favorable action is respectfully requested.

In the Office Action, the following actions were taken or matters were raised:

Double Patenting

Claims 7-18 and 26-32 stand provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-13 of co-pending Application Serial No. 13/769,593. Given that the above-referenced double patenting rejection is provisional, Applicant respectfully submits that upon the allowance/issuance of either the instant Application and/or the 13/769,593 Application, Applicant will address any non-provisional double patenting rejection maintained by the Examiner.

35 U.S.C. § 102 Rejections

Claims 7, 8, 13 and 14 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by W.O. 2013/002777 of Hsu (hereinafter "Hsu"). Claims 7, 8, 13 and 14 stand rejected under 35 U.S.C. § 102(e) as allegedly being anticipated by U.S. Patent Publication No. 2012/0216135 of Wong et al. (hereinafter "Wong"). Applicant respectfully traverses these rejections, and the assertions and determinations therein, for at least the following reasons.

Of the rejected claims, Claims 7 and 13 are independent. Applicant respectfully submits that each of independent Claims 7 and 13 is patentable over the cited references. For example,

independent Claim 7 recites "a first machine having a running instance of an operating system and an application," "a management console operable to manage **migrating the running operating system and application from the first machine to a second machine**" and that the management console is operable to "display an interface comprising a plurality of selectable **mobility settings**, each mobility setting corresponding to a . . . resource allocation **to utilize for the migration**." (emphasis added). Applicant respectfully submits that the cited references does not disclose the above-referenced features of Claim 7.

Hsu reference

Hsu appears to disclose a system and method for migrating a running application from a first operating system environment to a second operating system environment. (Hsu, abstract). Hsu appears to disclose that an operating system container (205-1, 205-2) may be created as a virtualized operating system based on an already existing operating system 115 on a first system. (Hsu, paragraph 0030). Hsu appears to indicate that the operating system containers (205-1, 205-2) do not execute separate instances of the operating system kernel but, instead, obtain functionality implemented by the already existing operating system 115. (Hsu, paragraph 0030). Each application may be assigned to one of the containers (205-1, 205-2) based on input by a user or automatically by the operating system 115 based on heuristics or rules. (Hsu, paragraphs 0031 and 032). For example, the operating system 115 may determine an amount of processing resources utilized by each of the running applications and the amount of available resources in external computing systems. (Hsu, paragraph 0032). The operating system 115 may then match applications and operating system containers to the external computing systems. (Hsu, paragraph 0032).The operating system containers (205-1, 205-2) may then be migrated from their current/first computing system to another/second computing system. (Hsu, paragraph 0037). For example, the second computing system may implement a second operating system environment with a second set of hardware executing a second operating system 415. (Hsu, paragraph 0037).

Hsu does not appear to disclose the migration of a running instance of an operating system and an application.

In Hsu, the running instance of the operating system 115 of Hsu is not migrated. Hsu appears to indicate that only the running applications are migrated. For example, Hsu appears to indicate that the functionality of the containers (205-1, 205-2) is implemented by the already

existing operating system 115. However, the operating system 115 of the first computing system remains on the first computing system. The second computing system has its own running instance of an operating system (e.g., operating system 415-1 on system 405-1, and operating system 415-2 on system 405-2). Thus, Hsu does not appear to disclose the migration of a running instance of an operating system.

The interface of Hsu does not appear to include selectable mobility settings defining resources to utilize for the migration.

In Hsu, an interface appears to be presented to a user to enable the user to create a container (205-1, 205-2) and assign certain applications to certain containers. (Hsu, paragraphs 0031 and 0043). However, this interface does not appear to include selectable mobility settings defining resources to utilize for the migration. Although the container (205-1, 205-2) may be created to accommodate processing resources **currently being utilized** by an application, the interface does not appear to enable a selection of a setting to define a resource allocation to use **for the migration**.

Accordingly, for at least these reasons, Hsu does not appear to disclose each and every limitation recited by Claim 7. Therefore, Applicant respectfully requests that the rejection of Claim 7 be withdrawn.

Independent Claim 13 recites computer readable program code configured to "receive a request to **migrate a logical partition (LPAR) running an instance of an operating system and an application from a first machine to a second machine**," "display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, **each mobility setting corresponding to a . . . resource allocation to utilize for the migration**," "receive a selection of at least one mobility setting," and "**migrate the LPAR from the first machine to the second machine utilizing the . . . resource allocation as set by the selected mobility setting**." (emphasis added). At least for the reasons indicated above in connection with independent Claim 7, Applicant respectfully submits that the cited reference does not disclose each and every limitation recited by Claim 13. Accordingly, Applicant respectfully requests that the rejection of Claim 13 be withdrawn.

Wong reference

Wong appears to disclose an interface for managing a virtualized computing environment. (Wong, abstract). The interface of Wong appears to enable the management of the virtualization infrastructure, including managing the host computers, the virtual machines running within each host computer, provisioning, migration, resource allocations, etc. (Wong, paragraph 0048). Wong appears to indicate that the infrastructure may include a number of virtual machines each with a guest operating system, and that a user of the interface may wish to migrate a virtual machine from one host computer to a different host computer. (Wong, paragraphs 0022 and 0043). Wong appears to disclose that the user may click on an object representing the virtual machine and drag the object to another portion of a visual hierarchy representing a second host computer. (Wong, paragraph 0043).

The interface of Wong does not appear to enable a user to select a setting for a resource allocation to apply for a virtual machine migration.

Although Wong appears to indicate that the interface of Wong may be used to manage a virtual machine (e.g., provisioning a machine, allocating resources to the machine, and even migrating the machine), there appears to be **no disclosure in Wong that enables a selection of a particular resource allocation that will be used for the migration of the machine from one host to another host**. For example, according to Wong, the interface of Wong enables a user to essentially drag and drop a visual representation of a virtual machine between different hosts to facilitate a migration of the virtual machine. However, this does not equate to a selection of a setting for a resource allocation to apply for a migration of the virtual machine. **Nor does a resource allocation for running the virtual machine equate to a resource allocation for migrating the virtual machine**. Thus, Applicant respectfully submits that Wong does not appear to disclose a management console operable to "display an interface comprising a plurality of selectable **mobility settings**, each mobility setting corresponding to a . . . resource allocation **to utilize for the migration**" as recited by Claim 7. (emphasis added).

Accordingly, for at least these reasons, Wong does not appear to disclose each and every limitation recited by Claim 7. Therefore, Applicant respectfully requests that the rejection of Claim 7 be withdrawn.

Independent Claim 13 also recites computer readable program code configured to "receive a request to **migrate a logical partition (LPAR) running an instance of an operating system and an application from a first machine to a second machine**," "display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, **each mobility setting corresponding to a . . . resource allocation to utilize for the migration**," "receive a selection of at least one mobility setting," and "**migrate the LPAR from the first machine to the second machine utilizing the . . . resource allocation as set by the selected mobility setting**." (emphasis added). At least for the reasons indicated above in connection with independent Claim 7, Applicant respectfully submits that the cited reference does not disclose each and every limitation recited by Claim 13. Accordingly, Applicant respectfully requests that the rejection of Claim 13 be withdrawn.

Claims 8 and 14 that depend respectively from independent Claims 7 and 13 are also not anticipated by Hsu or Wong at least because they incorporate the limitations of respective Claims 7 and 13 and also add additional elements that further distinguish Hsu and Wong. Therefore, Applicant respectfully requests that the rejection of Claims 7, 8, 13 and 14 be withdrawn.

35 U.S.C. § 103 Rejections

Claims 9-12, 15-18 and 26-32 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Hsu in view of U.S. Patent Publication No. 2010/0205602 of Zedlewski et al. (hereinafter "Zedlewski"). Applicant respectfully traverses this rejection, and the assertions and determinations therein, for at least the following reasons.

Claims 9-12, 15-18 and 32 depend from respective independent Claims 7 and 13. As indicated above, Applicant respectfully submits that independent Claims 7 and 13 are patentable over the cited Hsu reference. Therefore, Claims 9-12, 15-18 and 32 that depend from respective independent Claims 7 and 13 are also patentable. Further, Zedlewski does not appear to remedy the deficiencies of Hsu indicated above. Therefore, Applicant respectfully requests that the rejection of Claims 9-12, 15-18 and 32 be withdrawn.

Of the remaining rejected claims, Claim 26 is independent. Applicant respectfully submits that independent Claim 26 is patentable over the cited references. For example, independent Claim 26 recites "a first machine having a plurality of logical partitions (LPARs)

each running an instance of an operating system and an application" and "a management console operable to manage migrating the plurality of LPARs from the first machine to a second machine," and where the management console further operable to "display an interface comprising a plurality of **selectable mobility settings**, **each mobility setting corresponding to a desired resource allocation to utilize for the migration**, wherein a **first mobility setting sets a first resource allocation to accommodate a desired rate of migration**, and wherein a **second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations**." (emphasis added). Applicant respectfully submits that the cited references do not disclose the above-referenced features of Claim 26.

Hsu is relied on in the Office Action as purportedly teaching the migration of a running instance of an operating system. (Office Action, pages 21 and 22). However, as indicated above in connection with independent Claim 7, Hsu does not migrate a running instance of an operating system.

Further, Hsu is relied on in the Office Action as purportedly teaching an interface having a plurality of mobility settings for a desired resource allocation to use for a migration. (Office Action, pages 21 and 22). However, as indicated above in connection with independent Claim 7, the interface of Hsu appears to enable a user to create a container with resources to accommodate a currently running application, but there appears to be no interface that enables the allocation of resources that will be used **for the migration**.

Additionally, Zedlewski appears to be relied upon to teach a mobility setting to set a resource allocation to accommodate a desired rate of LPAR migration and a desired concurrency of LPAR migrations. (Office Action, pages 22 and 23). Applicant respectfully disagrees.

Zedlewski appears to disclose that shared resources may be accessed by multiple applications, and that threads may be executed "concurrently." (Zedlewski, paragraphs 0007-0011). Zedlewski does not appear to have any disclosure even remotely related to a rate of LPAR migration or a concurrency of LPAR migrations. Although Zedlewski appears to disclose, and the Examiner appears to rely upon, concurrently executed threads, there appears to be no setting disclosed in Zedlewski, in an interface or otherwise, that enables a selection of a resource

allocation applied to a migration that accommodates a desired rate of LPAR migration or a concurrency of LPAR migrations from one machine to another machine.

Moreover, the Examiner appears to assert that "concurrency-based," interpreted broadly, means that any allocation may be concurrency-based (e.g., when nothing else is concurrently running/being processed, when x threads are being processed, etc.). (Office Action, pages 22 and 23). Applicant respectfully disagrees. The Examiner appears to view the "concurrency" limitation in isolation without evaluating the remainder of the claim. For example, Claim 26 recites "wherein a first mobility setting **sets a first resource allocation to accommodate a desired rate of migration**, and wherein a second mobility setting **sets a second resource allocation to accommodate a desired concurrency of LPAR migrations**." (emphasis added). Thus, the "concurrency" recited by Claim 26 is directed toward the concurrency of "LPAR migrations." Accordingly, Applicant respectfully submits that the Examiner's interpretation of Claim 26 is improper.

Thus, for at least these reasons, Applicant respectfully submits that the cited references fail to disclose the above-referenced limitations of Claim 26. Accordingly, Applicant respectfully requests that the rejection of Claim 26 be withdrawn.

Claims 27-31 depend from independent Claim 26. As indicated above, Applicant respectfully submits that independent Claim 26 is patentable over the cited references. Therefore, Claims 27-31 that depend from independent Claim 26 are also patentable. Therefore, Applicant respectfully requests that the rejection of Claims 27-31 be withdrawn.

CONCLUSION

Having now responded to each rejection set forth in the present Office Action, Applicant respectfully submits that all claims now pending are in condition for allowance and respectfully requests such allowance.

No fees are believed due with this Response. However, if Applicant has overlooked the need for any fee, the Commissioner is hereby authorized to charge any fees or credit any overpayments associated with this Response to the deposit account of IBM Corporation, Deposit Account No. 09-0447.

Respectfully submitted,

By: /James L. Baudino/

James L. Baudino
Reg. No. 43,486

Date: September 11, 2014

IBM Corporation (AUS)
c/o The Law Office of James Baudino, PLLC
2313 Roosevelt Drive
Suite A
Arlington, Texas 76016
Phone: 817-751-7140
Fax: 817-751-7141

REMARKS

Claims 7-18 were pending at the mailing of a final Office Action dated August 2, 2013. Reconsideration and allowance of the claims is respectfully requested.

In the Office Action, the following actions were taken or matters were raised:

Double Patenting

Claims 7-18 stand provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-13 of co-pending Application Serial No. 13/769,593. Given that the above-referenced double patenting rejection is provisional, Applicant respectfully submits that upon the allowance/issuance of either the instant Application and/or the 13/769,593 Application, Applicant will address any non-provisional double patenting rejection maintained by the Examiner.

35 U.S.C. § 102 Rejections

Claims 7-10 and 13-16 stand rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by U.S. Patent No. 6,538,669 of Lagueux et al. (hereinafter "Lagueux"). Applicant respectfully traverses this rejection, and the assertions and determinations therein, for at least the following reasons.

Of the rejected claims, Claims 7 and 13 are independent. Applicant respectfully submits that each of independent Claims 7 and 13 is patentable over the cited reference. For example, independent Claim 7 recites "a first machine having a running application" and "a management console operable to manage **migrating the running application from the first machine to a second machine**." (emphasis added). Applicant respectfully submits that the cited reference does not disclose the above-referenced features of Claim 7.

Lagueux appears to disclose a graphical user interface for the configuration of a storage system. (Lagueux, abstract). Lagueux appears to disclose that data can be moved from one storage array to another storage array. (Lagueux, column 8, line 67 to column 9, line 13). For example, Lagueux recites the following:

> [I]f storage 150 is a terabyte drive array and a new drive array that supports sixteen terabytes of storage is being brought onto the network, the ISAN server 102A can **migrate the data** to the new array without consuming processor time on the server 100A. Further, the mirroring drivers support intelligent read and write functionality that allow the storage 150 to be shifted to the new drive array without disrupting access to the data.
>
> Thus, **as data is moved** to the new storage array, **read and write requests for those portions of the data** will be directed to the new array while requests for other data portions will be directed to the old storage 150.

(Lagueux, column 8, line 67 to column 9, line 13) (emphasis added).

Lagueux does not appear to disclose the migration of a running application. In the Office Action, the Examiner appears to consider the movement of data from one storage array to another storage array as migrating a "running application." (Office Action, pages 3 and 4). Applicant respectfully disagrees. The migration of the storage 150 of Lagueux appears to be nothing more than the movement of data identified as storage 150 from one storage array to another storage array. The **movement of data in Lagueux is not the migration of a running application**.

The Examiner further states in the Office Action that an application is inherently processing/manipulating data, and that the "'migration of a running application' may, reasonably and broadly, be understood as migrating data **associated, at any level**, with a running application." (Office Action, page 3) (emphasis added). Applicant respectfully disagrees. There is a distinction between the data that may be manipulated (or caused to be manipulated) by a running application, and the running application itself. That is, a running application, such as an operating system or other type of application, comprises instructions that are being executed by a processor. The migration of a running application generally includes saving and relaying state information corresponding to the running application between machines. Thus, migrating a "running application" is not equivalent to the migration of "data" as asserted by the Examiner. Accordingly, the Examiner's interpretation of a "running application" is unreasonable and inconsistent with Applicant's specification. The Examiner also asserts that the term "application" is open-ended and not specific because Applicant's specification also refers to "program

instructions." Applicant respectfully submits that the Examiner interpretation appears to completely ignore the explicit limitation of a "running" application.

Thus, Applicant respectfully submits that Lagueux does not disclose the above-referenced limitation of Claim 7. Accordingly, for at least this reason, Lagueux does not anticipate Claim 7.

Further, in the Office Action, the Examiner appears to refer to a "persistent table module 1400" as corresponding to the "management console" recited by Claim 7. (Office Action, pages 10 and 11). Lagueux appears to disclose a persistent table module 1400 that includes a table data access manager 1402. (Lagueux, column 16, lines 713). Lagueux appears to disclose that the table data access manager 1402 and the table class manager 1405 may be used to configure a set of tables (e.g., a fiber channel port ID table, a LUN export table, a configuration template table, etc.). (Lagueux, column 16, lines 7-25). Thus, **the persistent table module 1400 of Lagueux does not appear to be used to manage the migration of a running application. Nor does the persistent table module 1400 of Lagueux appear to display an interface comprising "selectable mobility settings" that correspond to a desired resource allocation to apply for migrating a running application**. Thus, for this reason also, Lagueux does not appear to disclose the limitations of Claim 7.

The Examiner also appears to rely on a front panel display (FPD) 220 of an ISAN server 102A of Lagueux and a "management interface 120" of Lagueux to correspond to the "management console" and "interface" recited by Claim 7. (Office Action, pages 10 and 11). The FPD appears to include a user interface that supports status displays, configuration display and management, and other management functions. (Lagueux, column 10, lines 21-28). The interface 120 of Lagueux appears to include interfaces for "**rules based management of the system including: scheduling, or process orchestration; monitoring the system; informed consent management; and handling system processes and events**." (Lagueux, column 7, lines 20-30) (emphasis added). The mere recitation of "scheduling, or process orchestration" or "handling system processes," without more, does not read on the limitation of an interface displaying "selectable mobility settings" to utilize for migrating a running application." **Lagueux does not appear to provide any teaching that the FPD or interface of Lagueux displays "selectable**

mobility settings" each corresponding to "a desired resource allocation to utilize" for the migration of a running application.

Additionally, Lagueux does not appear to disclose the display or the receipt of a selection of a "mobility setting" that corresponds to "a desired resource allocation" to utilize for migrating a running application from one machine to another machine. In the Office action, the Examiner appears to rely on column 18, lines 51-61, as disclosing the above-referenced limitations recited by Claim 7. (Office Action, pages 11 and 12). Applicant respectfully disagrees.

Lagueux appears to disclose a representative ISAN server 102 (figure 2) that may have a number of connection options (e.g., serial 140, front panel 142, Ethernet 144, and network interface card (NIC) 146). (Lagueux, figure 2, column 6, lines 39-58). The Examiner appears to rely on **an example storage transaction** that, in this particular example, is coming over the network interface 146. (Office Action, page 11 (referring to Lagueux, column 18, lines 51-61)). In this particular example, Lagueux appears to disclose that, for a storage transaction arriving over the network interface, hardware device driver receives the transaction and dispatches it to an appropriate virtual device based on a protocol. (Lagueux, column 18, lines 51-61). Neither in the portion of Lagueux referenced by the Examiner, nor elsewhere in Lagueux, does there appear to be any disclosure of the display or the receipt of a selection of a "mobility setting" that corresponds to "a desired resource allocation" to utilize for migrating a running application from one machine to another machine as recited by Claim 7. Therefore, for at least this reason also, Lagueux does not appear to anticipate Claim 7.

Accordingly, for at least these reasons, Lagueux does not appear to disclose each and every limitation recited by Claim 7. Therefore, Applicant respectfully requests that the rejection of Claim 7 be withdrawn.

Independent Claim 13, as amended, recites computer readable program code configured to "receive a request to **migrate a logical partition (LPAR) running an instance of an operating system and an application from a first machine to a second machine**," "display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, **each mobility setting corresponding to a desired resource allocation to utilize for the migration**," "receive a selection of at least one mobility setting," and "**migrate the LPAR**

from the first machine to the second machine utilizing resources as set by the selected mobility setting." (emphasis added). At least for the reasons indicated above in connection with independent Claim 7, Applicant respectfully submits that the cited reference does not disclose each and every limitation recited by Claim 13. Moreover, Lagueux does not disclose the migration of a logical partition including a running instance of an operating system and an application. Accordingly, Applicant respectfully requests that the rejection of Claim 13 be withdrawn.

Claims 8-10 and 14-16 that depend respectively from independent Claims 7 and 13 are also not anticipated by Lagueux at least because they incorporate the limitations of respective Claims 7 and 13 and also add additional elements that further distinguish Lagueux. Therefore, Applicant respectfully requests that the rejection of Claims 7-10 and 13-16 be withdrawn.

35 U.S.C. § 103 Rejections

Claims 11, 12, 17 and 18 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Lagueux in view of U.S. Patent Publication No. 2011/0040725 of Sugimoto (hereinafter "Sugimoto"). Applicant respectfully traverses this rejection and the assertions and determinations therein, for at least the following reasons.

Claims 11, 12, 17 and 18 depend from respective independent Claims 7 and 13. As indicated above, Applicant respectfully submits that independent Claims 7 and 13 are patentable over the cited Lagueux reference. Therefore, Claims 11, 12, 17 and 18 that depend from respective independent Claims 7 and 13 are also patentable. Further, Sugimoto does not appear to remedy the deficiencies of Lagueux indicated above. Therefore, Applicant respectfully requests that the rejection of Claims 11, 12, 17 and 18 be withdrawn.

New Claims

Applicant adds new Claims 26-32. Of the new claims, Claim 26 is independent, Claims 27-31 depend from new Claim 26, and new Claim 32 depends from Claim 7. Applicant respectfully submits that each of new Claims 26-32 is patentable over the cited references.

For example, new Claim 32 recites that "each selectable mobility setting sets an amount of memory **on the first and second machines** **to allocate to the migration** and **a quantity of**

threads to use **on the first and second machines** **for the migration**" and "an allocation module configured to **determine availability** of the amount of memory and the quantity of threads **on the first and second machines** for the migration based on the selected mobility setting and, in response to determining an **unavailability of the amount of memory or the quantity of threads on either the first or second machines** for the migration, negotiate a balance of memory and threads to use on the first and second machines to use for the migration." (emphasis added).

The cited references do not disclose at least the above-referenced limitations of Claim 7. For example, even if the migrating of "data" in Lagueux is considered to be the migration of a "running application," which Applicant submits is not the case, Lagueux does not determine "an amount of memory" to use on both the source and destination volumes or the "quantity of threads to use" for the migration where the amount of **memory on the source** or the destination may be **unavailable** for the migration. That is, the amount of memory the data in Lagueux is occupying in the source volume cannot also be the memory used for the migration because such memory amount may be "unavailable" on the source for the migration. Also, neither Lagueux nor Sugimoto teaches determining the quantity and availability of threads on both the source and destination machines for a migration. Thus, Claim 7 is clearly patentable over the cited references.

CONCLUSION

Having now responded to each rejection set forth in the present Office Action, Applicant respectfully submits that all claims now pending are in condition for allowance and respectfully requests such allowance.

An RCE filing fee of $1200.00 is believed due with this RCE. The Commissioner is hereby authorized to charge $1200.00 for the RCE filing fee to the deposit account of IBM Corporation, Deposit Account No. 09-0447. If Applicant has miscalculated the fee due with this RCE or overlooked the need for any other fee, the Commissioner is hereby authorized to charge any fees or credit any overpayments associated with this RCE to the deposit account of IBM Corporation, Deposit Account No. 09-0447.

Respectfully submitted,

By: /James L. Baudino/

James L. Baudino
Reg. No. 43,486

Date: November 4, 2013

IBM Corporation (AUS)
c/o The Law Office of James Baudino, PLLC
600 Six Flags Drive
Suite 400
Arlington, Texas 76011
Phone: 817-751-7140
Fax: 817-751-7141

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/671,422	11/07/2012	Maria Garza	AUS920120309US1	9343

77351 7590 06/12/2014

IBM CORP. (AUS)
C/O THE LAW OFFICE OF JAMES BAUDINO, PLLC
2313 ROOSEVELT DRIVE
SUITE A
ARLINGTON, TX 76016

EXAMINER	
IPAKCHI, MARYAM M	
ART UNIT	**PAPER NUMBER**
2171	
MAIL DATE	**DELIVERY MODE**
06/12/2014	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.	Applicant(s)		
13/671,422	GARZA ET AL.		
Examiner	**Art Unit**	**AIA (First Inventor to File) Status**	
MARYAM IPAKCHI	2171	No	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11/4/2013*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *7-18 and 26-32* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *7-18 and 26-32* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

This action is responsive to Applicants' Amendment with RCE filed on November 4, 2013 by which claim 13 was amended and claims 26-32 were added.

Claims 7-18 and 26-32 are pending, of which claims 7, 13, and 26 are independent. Claims 1-6 and 19-25 are cancelled.

The application was filed in the US Patent Office on November 7, 2012 and claims no domestic benefit or foreign priority.

The application is currently assigned to International Business Machines Corporation.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on November 7, 2012 has been considered by the Examiner.

Response to Arguments

Applicant's arguments filed in the Amendment with RCE filed on November 4, 2013 have been fully considered but are moot in view of the following rejections set forth below.

The provisional obviousness-type double patenting rejection is maintained as all the co-pending applications and claims are still pending and none has yet reached a final disposition.

It is noted that, during examination, a claim must be given its broadest reasonable interpretation consistent with the specification. Under a broadest reasonable interpretation, *words of the claim must be given their plain meaning,* unless such meaning is *inconsistent* with the specification. M.P.E.P. 2173.01(I). It is respectfully submitted that each claim is to be interpreted based on the *language of the claim itself*, so long as that interpretation is consistent with the specification. Further, "though understanding of the claim language may be aided by explanations contained in the written description, it is important not to import into a claim limitations that are not part of the claim. For example, a particular embodiment appearing in the written description may not be read into a claim when the claim language is broader than the embodiment." M.P.E.P. 2111.01(II).

It is noted that care be taken such that the claims themselves explicitly recite all the claimed elements relied upon in overcoming the rejections set forth herein. That is, for any additional limitations discussed in the specification to be considered, the claims should be amended such that the limitations are explicitly recited in the claims themselves. Appropriate consideration of each and every feature of the claims has been made.

More particularly, in the following detailed rejections, the Examiner provides various "examiner notes" regarding the reasonable broad interpretation of claimed features. It is noted, e.g., that features in paragraph [0041] of the published originally filed specification (US 2014/0129958) if explicitly incorporated into the claim language, may be helpful in expediting prosecution.

Further, in the interests of compact and expedited prosecution, examiner also notes the alternative rejection of some of the claims with regard to Wong et al.

Applicants' representative is welcome and encouraged to contact the examiner (Maryam Ipakchi) at 571-27-3237 (tel) or 571-270-4237 (direct fax) to further discuss the claims, proposed amendments thereto, and/or the applied art in order to help advance/expedite prosecution.

Double Patenting

It is noted that Applicants/Assignee have filed multiple related applications. Applicants/Assignee should, in good faith, proactively file respective Terminal Disclaimers for all such closely related and/or identical applications, as necessary.

Claims 7-18 and 26-32 of the instant application are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being anticipated by claims 1-20 of co-pending US Application No. 13/769,593.

The respective conflicting claims while not identical, are not patentably distinct from each other. Claims 7-18 and 26-32 of the instant application correspond to products and claims 1-20 of copending US Application No. 13/769,593 correspond to obvious methods of using the products at least as the claims correspond to product claims and method claims, respectively, and the product claims carry out the claimed methods.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented. The nonstatutory double patenting

rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory obviousness-type double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent either is shown to be commonly owned with this application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement.

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

US Patent Application No. 13/769,593 to Garza et al.	Instant Application
1. A method, comprising:	7. A system, comprising:

receiving a request to migrate a running instance of an operating system and an application from a first machine to a second machine; displaying an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings comprising at least one performance-based mobility setting and at least one concurrency-based mobility setting; receiving, via the interface, a selection of a mobility setting defining a resource allocation to utilize for the migration; and migrating the running instance of the operating system and the application from the first machine to the second machine utilizing resources as set by the selected mobility setting.	a first machine having a running application; a management console operable to manage migrating the running application from the first machine to a second machine, the management console further operable to: display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration; receive, via the interface, a selection of a mobility setting to apply for migrating the running application; and initiate migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting.
2. The method of Claim 1, further comprising negotiating a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.	8. The system of Claim 7, further comprising at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.
3. The method of Claim 1, further comprising identifying a memory resource allocation for the migration based on the selected mobility setting.	10. The system of Claim 7, further comprising an allocation module operable to identify a memory resource allocation for the migration based on the selected mobility setting.
4. The method of Claim 3, further comprising determining processor utilization for the migration based on the selected mobility setting.	11. The system of Claim 10, wherein the allocation module is operable to identify a processor resource allocation for the migration based on the selected mobility setting.
5. The method of Claim 1, further comprising: allocating greater memory resources for the performance-based mobility setting than for the concurrency-based mobility setting; and allocating a greater quantity of threads for the	12. The system of Claim 7, wherein the interface is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a

performance-based mobility setting than for the concurrency-based mobility setting for managing the memory resources.	concurrency-based resource allocation.
6. The method of Claim 1, further comprising automatically overriding the mobility setting in response to identifying unavailable resources corresponding to a resource allocation indicated by the mobility setting.	9. The system of Claim 8, wherein the VIOS partition is operable to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines.
7. A method, comprising: receiving a request to migrate a plurality of logical partitions from a first machine to a second machine, each logical partition comprising a running instance of an operating system; displaying an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration; receiving, via the interface, a first mobility setting to apply to a first set of logical partitions of the plurality of logical partitions and a second mobility setting to apply to a second set of logical partitions of the plurality of logical partitions; and initiating migration of the first and second sets of logical partitions from the first machine to the second machine utilizing the resource allocations as set by the respective first and second mobility settings.	13. A computer program product for migration operation resource allocation, the computer program product comprising: a computer readable storage medium having computer readable program code embodied therewith, the computer readable program code comprising computer readable program code configured to: receive a request to migrate a logical partition (LPAR) running an instance of an operating system and an application from a first machine to a second machine; display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration; receive a selection of at least one mobility setting; and migrate the LPAR from the first machine to the second machine utilizing resources as set by the selected mobility setting.
8. The method of Claim 7, wherein the first mobility setting is a performance-based mobility setting and the second mobility setting is a concurrency-based mobility setting. 13. The method of Claim 8, further comprising: allocating greater memory resources for the first mobility setting than for the second	18. The computer program product of Claim 13, wherein the computer readable program code is configured to display the interface to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.

mobility setting; and allocating a greater quantity of threads for the first mobility setting than for the second mobility setting for managing the memory resources.	
9. The method of Claim 7, further comprising negotiating a balance of resource allocations between the first machine and the second machine based on the first and second mobility settings.	14. The computer program product of Claim 13, wherein the computer readable program code is configured to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting.
10. The method of Claim 9, further comprising automatically overriding the resource allocation indicated by either the first or second mobility settings in response to identifying unavailable resources on either the first or second machines.	15. The computer program product of Claim 14, wherein the computer readable program code is configured to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines.
11. The method of Claim 7, further comprising identifying a memory resource allocation for the	16. The computer program product of Claim 13, wherein the computer readable program code is configured to identify a memory resource allocation for the migration based on the selected mobility setting.
12. The method of Claim 11, further comprising identifying a processor resource allocation for the migration based on the first and second mobility settings.	17. The computer program product of Claim 16, wherein the computer readable program code is configured to identify a processor resource allocation for the migration based on the selected mobility setting.
14. A method, comprising: receiving a request to migrate from a first machine to a second machine a plurality of logical partitions (LPARs) each running an instance of an operating system and an application; displaying an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource	26. A system, comprising: a first machine having a plurality of logical partitions (LPARs) each running an instance of an operating system and an application; and a management console operable to manage migrating the plurality of LPARs from the first machine to a second machine, the management console further operable to:

allocation to utilize for the migration, wherein a first mobility setting sets a first resource allocation to accommodate a desired rate of migration, and wherein a second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations; receiving, via the interface, a selection of a mobility setting to apply for migrating the LPARs; and initiating migration of the LPARs from the first machine to the second machine utilizing resources as set by the selected mobility setting.	display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration, wherein a first mobility setting sets a first resource allocation to accommodate a desired rate of migration, and wherein a second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations; receive, via the interface, a selection of a mobility setting to apply for migrating the LPARs; and initiate migration of the LPARs from the first machine to the second machine utilizing resources as set by the selected mobility setting.
15. The method of Claim 14, further comprising setting an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration based on the selected mobility setting.	27. The system of Claim 26, wherein each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration.
16. The method of Claim 15, further comprising, in response to detecting a mismatch between the amount of memory on the first machine and the amount of memory on the second machine to allocate to the migration based on the selected mobility setting, negotiating a balance of resource allocation on the first and second machines based on available resources on the first and second machines.	28. The system of Claim 27, further comprising an allocation module configured to, in response to detecting a mismatch between the amount of memory on the first machine and the amount of memory on the second machine to allocate to the migration based on the selected mobility setting, negotiate a balance of resource allocation on the first and second machines based on available resources on the first and second machines.
17. The method of Claim 16, further comprising receiving a different selected mobility setting for each of the plurality of LPARs and automatically applying the respective selected mobility settings to the respective migrations of the LPARs.	29. The system of Claim 26, wherein the management console is operable to receive a different selected mobility setting for each of the plurality of LPARs and automatically apply the respective selected mobility settings to the respective migrations of the

	LPARs.
18. The method of Claim 17, further comprising setting a run time for the threads based on the selected mobility setting.	30. The system of Claim 27, wherein each selectable mobility setting sets a run time for the threads.
19. The method of Claim 18, further comprising evaluating available resources on the first and second machines based on the selected mobility setting.	31. The system of Claim 27, further comprising an allocation module configured to evaluate available resources on the first and second machines based on the selected mobility setting.
20. The method of Claim 1, wherein: each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration; and further comprising determining availability of the amount of memory and the quantity of threads on the first and second machines for the migration based on the selected mobility setting and, in response to determining an unavailability of the amount of memory or the quantity of threads on either the first or second machines for the migration, negotiate a balance of memory and threads to use on the first and second machines to use for the migration.	32. The system of Claim 7, wherein: each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration; and further comprising an allocation module configured to determine availability of the amount of memory and the quantity of threads on the first and second machines for the migration based on the selected mobility setting and, in response to determining an unavailability of the amount of memory or the quantity of threads on either the first or second machines for the migration, negotiate a balance of memory and threads to use on the first and second machines to use for the migration.

It is noted that at this stage in the prosecution of the related applications, as none of the claims have reached a final disposition, Applicant is respectfully reminded to take all necessary measures, e.g., amend the claims and/or file terminal disclaimers, as necessary, to preemptively avoid double patenting and to avoid any unnecessary delay

in prosecution/final disposition of any one or more of the applications as a result of such issues and/or the need to file a terminal disclaimer.

35 USC § 112(f)

The following is a quotation of 35 U.S.C. 112(f):

> (f) Element in Claim for a Combination. – An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

The following is a quotation of pre-AIA 35 U.S.C. 112, sixth paragraph:

> An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof.

Claim limitation “a management console” of each of claims 7 and 26 has been interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, because it uses use a generic placeholder “management console” coupled with functional language “operable to manage migrating the plurality of LPARs from the first machine to the second machine, the management console further operable to display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration, wherein a first mobility setting sets a first resource allocation to accommodate a desired rate of migration, and wherein a second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations; receive, via the interface, a selection of a mobility setting to apply for migrating the LPARs; and initiate migration of the LPARs from the first machine to the second machine utilizing resources as set by the selected

mobility setting" without reciting sufficient structure to achieve the function. Furthermore, the generic placeholder is not preceded by a structural modifier.

Similarly, the claim limitation "an allocation module" of each of claims 10, 11, 28, 31, and 32 has been interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, because it uses use a generic placeholder "allocation module" coupled with functional language and fails to recite any structure for achieving the recited function.

Since the claim limitations invoke 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, claims 7-12 and 26-32 have been interpreted to cover the corresponding structure described in the specification that achieves the claimed function, and equivalents thereof.

A review of the specification shows that the following appears to be the corresponding structure described in the specification for the 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph limitation: [0011] and [0038] of Applicants' published application.

If applicant wishes to provide further explanation or dispute the examiner's interpretation of the corresponding structure, applicant must identify the corresponding structure with reference to the specification by page and line number, and to the drawing, if any, by reference characters in response to this Office action.

If applicant does not intend to have the claim limitation(s) treated under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112 , sixth paragraph, applicant may amend the claim(s) so that it/they will clearly not invoke 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph, or present a sufficient showing that the claim recites/recite sufficient

structure, material, or acts for performing the claimed function to preclude application of 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph.

For more information, see MPEP § 2173 *et seq.* and *Supplementary Examination Guidelines for Determining Compliance With 35 U.S.C. 112 and for Treatment of Related Issues in Patent Applications*, 76 FR 7162, 7167 (Feb. 9, 2011).

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 7, 8, 13, 14 are rejected under 35 U.S.C. 102(e) as being anticipated by WO 2013/002777 to Hsu.

Regarding independent claim 7, Hsu teaches:

A system (Hsu, FIG. 1, [0026] … system 100)**, comprising:**
a first machine having a running application (Hsu, FIGS. 1, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system);
a management console operable to manage migrating the running application from the first machine to a second machine, the management console further operable to display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Hsu, FIGS. 1, 3, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0032]

assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface*);

receive, via the interface, a selection of a mobility setting to apply for migrating the running application (Hsu, FIGS. 1, 3, 4; [0031] each of the applications (105-1 to 105-3) may be assigned to one of the created containers … assignment based on input from a user of the computing system through a user interface); **and**

initiate migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting (Hsu, FIGS. 1-4; [0033] once applications have been assigned to respective containers, the operating system may transfer the applications to their respective containers; [0040] from perspective of a user or client of the applications migrated from the first computing system, it will appear as if the applications have been running continuously without any substantive change in functionality … as if they were running on the source computing system; [0043]).

Regarding dependent claims 8 and 14, Hsu teaches:

at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting (Hsu, FIG. 4; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface* … provides flexibility for users to selectively migrate running applications to external operating system environments in order to perform maintenance on the host operating system environment or *balance workloads among multiple computing systems*).

Regarding independent claim 13, Hsu teaches:

A computer program product for migration operation resource allocation, the computer program product comprising (Hsu, FIGS. 1, 4; [0026]);

a computer readable storage medium having computer readable program code embodied therewith (Hsu, FIGS. 1, 4; [0026])**, the computer readable program code comprising computer readable program code configured to:**

receive a request to migrate a logical partition (LPAR) running an instances of an operating system and an application from a first machine to a second machine (Hsu, FIGS. 1, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface* … provides flexibility for users to selectively migrate running applications to external operating system environments in order to perform maintenance on the host operating system environment);

display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Hsu, FIGS. 1, 3, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0032] assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input toe the host computing system through a user interface*);

receive a selection of at least one mobility setting (Hsu, FIGS. 1, 3, 4; [0031] each of the applications (105-1 to 105-3) may be assigned to one of the created containers … assignment based on input from a user of the computing system through a user interface); **and**

migrate the LPAR from the first machine to the second machine utilizing resources as set by the selected mobility setting (Hsu, FIGS. 1-4; [0033] once applications have been assigned to respective containers, the operating system may transfer the applications to their respective containers; [0040] from perspective of a user or client of the applications migrated from the first computing system, it will appear as if the applications have been running continuously without any substantive change in functionality … as if they were running on the source computing system; [0043]).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 9-12, 15-18 and 26-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over WO 2013/002777 to Hsu in view of US Patent Publication 2010/0205602 to Zedlewski et al.

Regarding dependent claims 9 and 15, Hsu may fail to explicitly teach each and every feature of:

the VIOS partition (cl. 9)/computer program readable code (cl. 15) is operable to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines.

Zedlewski et al. teach:

the VIOS partition (cl. 9)/**computer program readable code** (cl. 15) **is operable to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines** (Zedlewski et al., [0009]-[0011], [0018] … at any time, the operating system can preempt a thread and force it to give up the CPU on which it is running in order to run another thread, [0021], [0031]-[0034] … *rescheduling* may be triggered according to rules programmed into the schedule, according to user-input parameters or both, or disabled altogether ... one measure of anti-cooperative execution behavior will be violation of user-specified thread performance requirement; FIGS. 5-6 … alternative scheduling option, [0053] … scheduler 610 may deal with the situations according to one or more options … conditions that trigger intervention may be user-selected or pre-set; …. Examiner notes claim language does not restrict basis of/interpretation of "unavailable resources" such that it the resource may be unavailable for any reason, i.e., error or physically unavailable).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). As both Hsu and Zedlewski pertain to efficient and effective data management and transfer mechanisms, it would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claims 10 and 16, Hsu may fail to explicitly teach each and every feature of:

an allocation module (cl. 10)/computer program readable code (cl. 16) operable to identify a memory resource allocation for the migration based on the selected mobility setting.

Zedlewski et al. teach:

an allocation module (cl. 10)/**computer program readable code** (cl. 16) **operable to identify a memory resource allocation for the migration based on the selected mobility setting** (Zedlewski et al., ., [0009]-[0011] … concurrent execution …; [0018], [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the

scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). As both Hsu and Zedlewski pertain to efficient and effective data management and transfer mechanisms, it would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claims 11 and 17, Hsu may fail to explicitly teach each and every feature of:

the allocation module (cl. 11)/computer program readable code (cl. 17) is operable to identify a processor resource allocation for the migration based on the selected mobility setting.

Zedlewski et al. teach:

the allocation module (cl. 11)/computer program readable code (cl. 17) **is operable to identify a processor resource allocation for the migration based on the selected mobility setting** (Zedlewski et al., [0009]-[0011] … concurrent execution … ; [0018], [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and

storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state; examiner notes claim language fails to provide any restrictive/condition regarding interpretation “processor resource allocation” or “selected mobility setting”).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claims 12 and 18, Hsu may fail to explicitly teach each and every feature of:

the interface (cl. 12)/computer program readable code (cl. 18) is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.

Zedlewski et al. teach:

the interface (cl. 12)/**computer program readable code** (cl. 18) **is configured to indicate a first mobility setting corresponding to a performance-based**

resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation (Zedlewski et al., [0009]-[0011] … concurrent execution …; [0018], [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … various settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state; [0096] e.g., users can select from choices for each VM ... any sharing, no sharing or internal sharing only .. default setting; Examiner notes that claim language does not specify any definition/restriction/conditions on the interpretation of the first and the second "mobility setting" as any setting can be interpreted as being performance based as the claim provides no measure regarding “performance based” … good performance, bad performance, etc., and similarly the claim language fails to set forth any definition for “concurrency-based” resource allocation and thus any allocation may be concurrency based as the claim language fails to provide any basis/condition for concurrency based, e.g., when nothing else is concurrently running/being processed, when x threads are being processed, etc.).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding independent claim 26, Hsu teaches:

A system, comprising: a first machine having a plurality of logical partitions (LPARs) each running an instance of an operating system and an application (Hsu, FIGS. 1, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface* … provides flexibility for users to selectively migrate running applications to external operating system environments in order to perform maintenance on the host operating system environment);

a management console operable to manage migrating the plurality of LPARs from the first machine to a second machine, the management console further operable to (Hsu, FIG. 5 … process management subsystem 535, process containerization module 530, live migration orchestrator 535):

display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Hsu, FIGS. 1, 3, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0032] assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input toe the host computing system through a user interface*); [0032], [0043] … examiner notes claim language does not define “desired” such that any setting may be deemed “desired”)**,**

receive, via the interface, a selection of a mobility setting to apply for migrating the LPARs (Hsu, FIGS. 1, 3, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0032] assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems; [0043]

operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input toe the host computing system through a user interface*); [0032], [0043])**; and**

initiate migration of the LPARs from the first machine to the second machine utilizing resources as set by the selected mobility setting (Hsu, FIGS. 1-4; [0033] once applications have been assigned to respective containers, the operating system may transfer the applications to their respective containers; [0040] from perspective of a user or client of the applications migrated from the first computing system, it will appear as if the applications have been running continuously without any substantive change in functionality … as if they were running on the source computing system; [0043]).

While Hsu teaches while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

wherein a first mobility setting sets a first resource allocation to accommodate a desired rate of migration, and wherein a second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations.

Zedlewski et al. teach:

wherein a first mobility setting sets a first resource allocation to accommodate a desired rate of migration, and wherein a second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations (Zedlewski et al., [0009]-[0011] … concurrent execution …; [0018], [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … various settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state; [0096] e.g., users can select from choices for each VM ... any sharing, no sharing or internal sharing only .. default setting; Examiner notes that claim language does not specify any definition/restriction/conditions on the interpretation of the “concurrency-based” resource allocation and thus any allocation may be concurrency based as the claim language fails to provide any basis/condition for

concurrency based, e.g., when nothing else is concurrently running/being processed, when x threads are being processed, etc.).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 27, while Hsu teaches that assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems and that the operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface* (Hsu, [0032], [0043]), Hsu et al. may fail to explicitly teach each and every feature of:

each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration.

Zedlewski et al. teach:

each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration (Zedlewski et al., [0009]-[0010] … standing goal of all computer design is to enable applications to run as fast and as efficiently as possible, even when sharing system resources and one way to accomplish this is efficient schedule for executing threads; [0018], [0074], [0094]-[0099] allocate suitable resources … threads count exceeds).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 28, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

an allocation module configured to, in response to detecting a mismatch between the amount of memory on the first machine and the amount of memory on the second machine to allocate to the migration based on the selected mobility setting, negotiate a balance of resource allocation on the first and second machines based on available resources on the first and second machines.

Zedlewski et al. teach:

an allocation module configured to, in response to detecting a mismatch between the amount of memory on the first machine and the amount of memory on the second machine to allocate to the migration based on the selected mobility setting, negotiate a balance of resource allocation on the first and second machines based on available resources on the first and second machines ([0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 29, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

the management console is operable to receive a different selected mobility setting for each of the plurality of LPARs and automatically apply the respective selected mobility settings to the respective migrations of the LPARs.

Zedlewski et al. teach:

the management console is operable to receive a different selected mobility setting for each of the plurality of LPARs and automatically apply the respective selected mobility settings to the respective migrations of the LPARs (Zedlewski et al., [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 30, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

each selectable mobility setting sets a run time for the threads.

Zedlewski et al. teach:

each selectable mobility setting sets a run time for the threads (Zedlewski et al., [0021] … known schedulers may adjust the amount of execution time allocated to each of a set of runnable threads. … Examiner further notes, however, that the claim language does not distinguish features of run time such that all the run times may be the same and may even be a default max time after which, e.g., an error/failure may be identified).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 31, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

an allocation module configured to evaluate available resources on the first and second machines based on the selected mobility setting.

Zedlewski et al. teach:

an allocation module configured to evaluate available resources on the first and second machines based on the selected mobility setting (Zedlewski et al., [0018] … the OS designates which software threads the logical processors are to execute and can also issue commands to cause an idle logical processor to be put in a halt state such that execution resources are made available for use by any remaining logical processors (examiner notes such a determination to cause an idle logical processor to be put in a halt state inherently requires an 'evaluation' of available resources); [0031] … rescheduling … upon sensing the rescheduling condition … ; … Examiner further notes, however, that the claim language does not set forth features of ‘evaluate’ such that any processing may be deemed to be evaluating available resources between the identified machines).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 32, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration; and further comprising an allocation module configured to determine availability of the amount of memory and the quantity of threads on the first and second machines for the migration based on the selected mobility setting and in response to determining an unavailability of the amount of memory or the quantity of threads on either the first or second machines for the migration, negotiate a balance of memory and threads to use on the first and second machines to use for the migration.

Zedlewski et al. teach:

each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration; and further comprising an allocation module configured to determine availability of the amount of memory and the quantity of threads on the first and second machines for the migration based on the selected mobility setting and (Zedlewski et al., [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0099] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state; thread count exceeds).

in response to determining an unavailability of the amount of memory or the quantity of threads on either the first or second machines for the migration, negotiate a balance of memory and threads to use on the first and second machines to use for the migration (Zedlewski et al., [0018] … the OS designates which software threads the logical processors are to execute and can also issue commands to cause an idle logical processor to be put in a halt state such that execution resources are made available for use by any remaining logical processors (examiner notes such a determination to cause an idle logical processor to be put in a

halt state inherently requires an 'evaluation' of available resources); [0031] … rescheduling … upon sensing the rescheduling condition … ; … Examiner further notes, however, that the claim language does not set forth features of 'negotiate' such that any allocation/processing may be deemed to be negotiation available resources among the machines).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 7, 8, 13 and 14 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 2012/0216135 to Wong et al.

Regarding claims 7 and 13, Wong et al. teach:

A system (Wong et al., FIGS. 1A-1B; system 100)**, comprising:**

a first machine having a running application (Wong et al., FIG. 1A, 1B, one of the host computers 104; [0017]-[0018] … host computers 104 … virtual machines running within each host computer 104);

a management console operable to manage migrating the running application from the first machine to a second machine, the management console further operable to display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Wong et al., FIGS. 1A-3, 7A, 7B; [0018]-[0019] ...virtual machine management center 102; … VM management center 102 manages the virtual infrastructures, including managing the host computers 104, the virtual machines running within each host computer 104, provisioning, migration, resource allocations; [0021] virtual machine monitor 125, which is a software interface layer that enables sharing of the hardware resources of host computer 104 by the virtual machines; [0026] ... allows disk images of virtual machines to be migrated between storage arrays without interrupting the virtual machine whose disk image is being migrated or any applications running inside it; [0032] GUI, command line interface; [0042]-[0043] method 510 at step 502 … graphical user interface includes receiving a mouse click on one or more objects in the graphical user interface, i.e., to select one the one or more objects. … the VI client can include a graphical user interface that allows a user manually migrate a disk image of a virtual machine between storage arrays; [0045] “Get VM1” … “Move”);

receive, via the interface, a selection of a mobility setting to apply for migrating the running application (Wong et al., FIGS. 1A-3, 7A, 7B; [0018]-[0019] ...virtual machine management center 102; … VM management center 102 manages the virtual infrastructures, including managing the host computers 104, the virtual machines running within each host computer 104, provisioning, migration, resource allocations; [0021] virtual machine monitor 125, which is a software interface layer that enables sharing of the hardware resources of host computer 104 by the virtual machines; [0026] ... allows disk images of virtual machines to be migrated between storage arrays without interrupting the virtual machine whose disk image is being migrated or any applications running inside it; [0032] GUI, command line interface;

[0042]-[0045] method 510 at step 502 … graphical user interface includes receiving a mouse click on one or more objects in the graphical user interface, i.e., to select one the one or more objects … the VI client can include a graphical user interface that allows a user manually migrate a disk image of a virtual machine between storage arrays); **and**

initiate migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting (Wong et al., FIGS. 1A-3, 7A, 7B; [0018]-[0019] ...virtual machine management center 102; … VM management center 102 manages the virtual infrastructures, including managing the host computers 104, the virtual machines running within each host computer 104, provisioning, migration, resource allocations; [0026] ... allows disk images of virtual machines to be migrated between storage arrays without interrupting the virtual machine whose disk image is being migrated or any applications running inside it; [0032] GUI, command line interface; [0042]-[0045] to accomplish the task of migrating the first VM the first host computer to the second host computer, the user may click on an object representing the first VM and drag the object to another portion of the visual hierarchy. In some embodiments, the user interface can display only technically feasible or recommended destinations for such a VM migration. … the VI client can include a graphical user interface that allows a user to manually migrate disk images of a virtual machine between storage arrays, as described in FIG. 1B).

Regarding claims 8 and 14, Wong et al. teach:

at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting (Wong et al., [0017] virtualization software partitions a physical host computer 104 into multiple secure and portable virtual machines that run on the same physical server).

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Examiner has cited particular columns and line numbers (or paragraphs) in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific imitations within the individual claim, other passages and figures may apply as well. More particularly, e.g., in the instances the Examiner has identified Figures of the applied prior art reference, it is understood that the corresponding portion of the written description describing the identified Figures is relied upon. It is respectfully requested from the Applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. **The entire reference(s) is/are to be considered to provide disclosure relating to the claimed invention.**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MARYAM IPAKCHI whose telephone number is (571)270-3237. The examiner can normally be reached on M-F 9-3:00EST. **Any interview requests should be made via an Interview Agenda faxed to the examiner at (571)270-4237. The Interview Agenda should include claims/items to be discussed and proposed dates/times.**

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt M. Kim can be reached on (571)272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MARYAM IPAKCHI/
Examiner
Art Unit 2171

/MATT KIM/
Supervisory Patent Examiner, Art Unit 2171

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/671,422	11/07/2012	Maria Garza	AUS920120309US1	9343

77351 7590 10/21/2014

IBM CORP. (AUS)
C/O THE LAW OFFICE OF JAMES BAUDINO, PLLC
2313 ROOSEVELT DRIVE
SUITE A
ARLINGTON, TX 76016

EXAMINER	
IPAKCHI, MARYAM M	
ART UNIT	**PAPER NUMBER**
2171	

MAIL DATE	DELIVERY MODE
10/21/2014	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.	Applicant(s)		
13/671,422	GARZA ET AL.		
Examiner	**Art Unit**	**AIA (First Inventor to File) Status**	
MARYAM IPAKCHI	2171	No	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *9/11/2014*.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5)☒ Claim(s) *7-18 and 26-32* is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6)☐ Claim(s) _____ is/are allowed.

7)☒ Claim(s) *7-18 and 26-32* is/are rejected.

8)☐ Claim(s) _____ is/are objected to.

9)☐ Claim(s) _____ are subject to restriction and/or election requirement.

* If any claims have been determined allowable, you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information, please see http://www.uspto.gov/patents/init_events/pph/index.jsp or send an inquiry to PPHfeedback@uspto.gov.

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

Certified copies:

a)☐ All b)☐ Some** c)☐ None of the:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date _____.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

4) ☐ Other: _____.

DETAILED ACTION

This action is responsive to Applicants' Amendment filed September 11, 2014 by which claims 7, 9, 13, and 15 were amended.

Claims 7-18 and 26-32 are pending, of which claims 7, 13, and 26 are independent. Claims 1-6 and 19-25 are cancelled.

The application was filed in the US Patent Office on November 7, 2012 and claims no domestic benefit or foreign priority.

The application is currently assigned to International Business Machines Corporation. The present application is being examined under the pre-AIA first to invent provisions.

Further to the Office Action mailed June 12, 2014, the claim limitation “a management console” of each of claims 7 and 26 and the “an allocation module” of each of claims 10, 11, 28, 31, and 32, as well as all the dependent claims thereof, have been interpreted under 35 U.S.C. 112(f) or pre-AIA 35 U.S.C. 112, sixth paragraph.

Information Disclosure Statement

The information disclosure statement (IDS) submitted on November 7, 2012 has been considered by the Examiner.

Response to Arguments

Applicants' amendments necessitated any new grounds of rejections presented below. Applicant's arguments filed in the response filed on September 11, 2014 have been fully considered but are not persuasive for at least the reasons set forth below.

The provisional obviousness-type double patenting rejection is held in abeyance as all the co-pending applications and claims are still pending and none has yet reached a final disposition.

It is noted that, during examination, a claim must be given its broadest reasonable interpretation consistent with the specification. Under a broadest reasonable interpretation, *words of the claim must be given their plain meaning,* unless such meaning is *inconsistent* with the specification. M.P.E.P. 2173.01(I). It is respectfully submitted that each claim is to be interpreted based on the *language of the claim itself*, so long as that interpretation is consistent with the specification. Further, "though understanding of the claim language may be aided by explanations contained in the written description, it is important not to import into a claim limitations that are not part of the claim. For example, a particular embodiment appearing in the written description may not be read into a claim when the claim language is broader than the embodiment." M.P.E.P. 2111.01(II).

It is further noted that care be taken such that the claims themselves explicitly recite all the claimed elements relied upon in overcoming the rejections set forth herein. That is, for any additional limitations discussed in the specification to

be considered, the claims should be amended such that the limitations are explicitly recited in the claims themselves. Appropriate consideration of each and every feature of the claims has been made.

At page 8 of Applicants' September 11, 2014 Amendment, it is stated that "Hsu does not appear to disclose the migration of a running instance of an operating system and an application" because the “running instance of the operating system 115 of Hsu is not migrated.” The Examiner respectfully disagrees for at least the reasons set forth below.

With regard to each of independent claims 7, 13, and 26, the examiner respectfully notes the claim language does not set forth specific/required features of the terms/phrase "a running instance of an operating system." Thus, e.g., the phrase may be reasonably and broadly interpreted as an ‘image’ or ‘mapping’ of any portion (e.g. partial) of an operating system or any process(es) therefrom, with or without the kernel itself, as an instance of an operating system that when, e.g., migrated to the new containers/systems (405-1, 405-2) appears to have been running continuously without any substantive change in functionality (Hsu, [0040]). The claim language does not explicitly require an instance of a complete operating system including the kernel. That is, without an explicit recitation of what constitutes a running instance of an operating system, the claim language “a running instance of an operating system and an application" may be broadly interpreted and be anticipated by, e.g., the applications (105-1, 105-2) and the operating system containers (205-1, 205-2) that refer to a virtual operating system defining an operating system environment, which refers to an instance

of an operating system kernel and a set of conditions under which an application may be executed using the operating system kernel (e.g., Hsu [0018]-[0020]) such that the containers and applications are migrated to a second computing system that when, e.g., migrated to the second containers/systems (405-1, 405-2) appear to have been running continuously without any substantive change in functionality (Hsu, [0040]).

Additionally, absent any specific language/limitations regarding the type/features of 'migration,' the term is subject to reasonable broad interpretation. It is noted that there are various known types of migration and ‘migration of running instance’ may be functionally performed via different virtualization approaches, e.g., operating system virtualization approach vs. operating system-level virtualization approach, etc. It is respectfully noted that absent a specific limiting language, the claim terms/phrases may be broadly interpreted.

At pages 9 and 10 of Applicants' September 11, 2014 Amendment, it is stated that "Hsu does not appear to include selectable mobility settings defining resources to utilize for the migration …. or to disclose any concurrency-based mobility setting" because the “interface appears to be presented to a user to enable the user to create a containers (205-1, 205-2) and assign certain applications to containers. ... Although the container (205-1, 205-2) may be created to accommodate processing resources currently being utilized by an application, the interface does not appear to enable a selection of a setting to define a resource allocation to use for the migration.” The Examiner respectfully disagrees for at least the reasons set forth below.

While it is thoroughly appreciated that an applicant may be his/her own lexicographer, as each of independent claims 7, 13, and 26 currently read, at least features of the claim terms "mobility setting" and "resource" are not defined/limited in any way. Respectfully, are the *applications/processes themselves not resources*? The claim language does not provide any limitation whatsoever on the features/types of the resources, e.g., hardware, software, virtual, etc. Are the applications/processes themselves not being migrated such that *assignment of the applications to particular containers via the user interface is not a mobility setting* defining which resource, e.g., application, is assigned to which container *used* ***during/for*** the migration? It is respectfully submitted that the allocated/assignment of applications is used during/for the migration at least in that it is known which application is migrated with which container, and the claim language does not prevent such a broad reasonable interpretation/analysis. Further, under broad reasonable interpretation of "concurrency-based mobility setting," two or more applications (e.g., resources) assigned/allocated to same container may be considered "concurrency-based mobility setting" or "two or more containers" being migrated from source to destination as part of a single migration. Additionally, it is respectfully noted that the claim language does not distinguish between "concurrency-based mobility setting" and "performance-based mobility setting" and thus, under broad reasonable interpretation they may even be the same thing or may be met by the same feature."

At pages 11 of Applicants' September 11, 2014 Amendment, it is stated that the same arguments provided for independent claim 7 similarly apply to

independent claim 13. Thus, for independent claim 13, the examiner similarly refers to the responses provided above with regard to Applicants' arguments regarding claim 1.

At page 10 of Applicants' September 11, 2014 Amendment, it is stated that Wong does not appear to enable a user to select a setting for a resource allocation to apply for a virtual machine migration. As discussed above, it is the language of the claim itself that is of importance/relevance, not features associated thereof read from the original specification. Further, the claim language itself does not specify any specific/required features of "resource allocation" or "setting" such that any allocation of resources and Wong explicitly teaches a user interface for managing resources in a virtualized computing environment as well as migration (Wong, [0005]-[0006], [0017]-[0018] … [0043] … interface displays interface ...allows a user to manually migrate images of a virtual machine between storage arrays).

At page 12 of Applicants' September 11, 2014 Amendment, it is stated that Zedlewski fails to teach first resource allocation to accommodate a desired rate of migration … second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations. Examiner notes that claim language does not specify any definition/restriction/conditions on the interpretation of the "concurrency-based" resource allocation and thus any allocation may be concurrency based as the claim language fails to provide any basis/condition for concurrency based, e.g., when nothing else is concurrently running/being processed, when x threads are being processed, as the claim language does not even require the

two settings to be different as a single setting may qualify as both and may be the same, particularly, when the claim language does not define/limit either and both are subject to broad reasonable interpretation. Additionally, it is respectfully noted what is a "desired rate of migration." At the very least, selecting applications assigned to a container, that affects the size of the container and the size of the container affects transfer rate, and the claim language does not prevent such an interpretation/analysis. Further, it is the combination of teachings of Zedlewski and Hsu in view of the claim language itself, i.e., no additional limitations may be read into the explicitly recited claim language. Additionally, the claim language does not positively tie "selection of a mobility setting" as necessarily being one of the "plurality of selectable mobility settings" as it merely says "via the interface, a selection of a mobility setting to apply" and does not say "via the interface, a selection of at least one of the plurality of selectable mobility settings."

Further, in the interests of compact and expedited prosecution, examiner also notes the newly cited references on the accompanying Notice of References Cited, particularly, the accompanying NPL references ("Live Migration of Virtual Machines" and "Migration without Virtualization") at least for support of the broad reasonable interpretation of the claim language. Additionally, the examiner notes US 6,226,734 to Kleinsorge et al., US 8,615,579 to Vincent et al., and US2012/0042034 to Goggin et al.

Applicants' representative is welcome and encouraged to contact the examiner (Maryam Ipakchi) at 571-270-3237 (tel) or 571-270-4237 (direct fax),

particularly, in view of the comments above, in order to further discuss the claims, proposed amendments thereto, and/or the applied art in order to help advance/expedite prosecution. The examiner welcomes any substantive discussion of proposed claim amendments in view of all of the art of record in advancing prosecution.

Double Patenting

It is noted that Applicants/Assignee have filed multiple related applications. Applicants/Assignee should, in good faith, proactively file respective Terminal Disclaimers for all such closely related and/or identical applications, as necessary.

As the current claims as well as those of pending applications are in flux, any potential provisional double-patenting rejection of the claims of the instant application with regard to, e.g., the claims of co-pending US Application No. 13/671,422 is held in abeyance until final disposition of any of the claims. Applicant is respectfully reminded to use consider all sets of pending claims to avoid any delay in prosecution as a result of potential double patenting issues.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before

the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 7, 8, 13, 14 are rejected under 35 U.S.C. 102(e) as being anticipated by WO 2013/002777 to Hsu.

Regarding independent claim 7, Hsu teaches:

A system (Hsu, FIG. 1, [0026] … system 100)**, comprising:**

a first machine having a running instance of an operating system and an application, the operating system and the application utilizing a first resource allocation to run on the first machine (Hsu, FIGS. 1, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system);

a management console operable to manage migrating the running operating system and application from the first machine to a second machine, the management console further operable to display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a second resource allocation to utilize for the migration (Hsu, FIGS. 1, 3, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0032] assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface*; examiner notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity or merely a virtual entity such that container allocation and/or application assignment may constitute a resource ... first resource allocation on first machine may be any allocation of memory (e.g., physical hardware 105) used by the application on the first machine 100);

receive, via the interface, a selection of a mobility setting to apply for migrating the running operating system and application (Hsu, FIGS. 1, 3, 4; [0031] each of the applications (105-1 to 105-3) may be assigned to one of the created containers … assignment based on input from a user of the computing system through

a user interface; examiner notes the assignment of applications to the containers, e.g., may be considered a mobility setting that is applied/used for the migration); **and**

initiate migration of the running operating system and application from the first machine to the second machine utilizing the second resource allocation as set by the selected mobility setting (Hsu, FIGS. 1-4; [0033] once applications have been assigned to respective containers, the operating system may transfer the applications to their respective containers; [0040] from perspective of a user or client of the applications migrated from the first computing system, it will appear as if the applications have been running continuously without any substantive change in functionality … as if they were running on the source computing system; [0043]; examiner notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity or merely a virtual entity such that container allocation and/or application assignment may constitute a resource ... first resource allocation on first machine may be any allocation of memory (e.g., physical hardware 105) used by the application on the first machine 100 to the second machine, e.g., FIG. 4, 405-1, 405-2 ...).

Regarding dependent claims 8 and 14, Hsu teaches:

at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting (Hsu, FIG. 4; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface* … provides flexibility for users to selectively migrate running applications to external operating system environments in order to perform maintenance on the host operating system environment or *balance workloads among multiple computing systems*).

Regarding independent claim 13, Hsu teaches:

A computer program product for migration operation resource allocation, the computer program product comprising (Hsu, FIGS. 1, 4; [0026]);

a computer readable storage medium having computer readable program code embodied therewith (Hsu, FIGS. 1, 4; [0026])**, the computer readable program code comprising computer readable program code configured to:**

receive a request to migrate a logical partition (LPAR) running an instances of an operating system and an application from a first machine to a second

machine, the LPAR utilizing a first resource allocation on the first machine (Hsu, FIGS. 1, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface* … provides flexibility for users to selectively migrate running applications to external operating system environments in order to perform maintenance on the host operating system environment);

display an adjustable resource allocation mobility setting interface indicating a plurality of mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Hsu, FIGS. 1, 3, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0032] assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface*; examiner further notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity or merely a virtual entity such that container allocation and/or application assignment may constitute a resource ... first resource allocation on first machine may be any allocation of memory (e.g., physical hardware) used by the application on the first machine 100 examiner notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity or merely a virtual entity such that container allocation and/or application assignment may constitute a resource ... first resource allocation on first machine may be any allocation of memory (e.g., physical hardware 105) used by the application on the first machine 100 to the second machine, e.g., FIG. 4, 405-1, 405-2 ...);

receive a selection of at least one mobility setting (Hsu, FIGS. 1, 3, 4; [0031] each of the applications (105-1 to 105-3) may be assigned to one of the created containers … assignment based on input from a user of the computing system through a user interface; examiner notes the assignment of applications to the containers, e.g., may be considered a mobility setting that is applied/used for the migration); **and**

migrate the LPAR from the first machine to the second machine utilizing the second resource allocation as set by the selected mobility setting (Hsu, FIGS.

1-4; [0033] once applications have been assigned to respective containers, the operating system may transfer the applications to their respective containers; [0040] from perspective of a user or client of the applications migrated from the first computing system, it will appear as if the applications have been running continuously without any substantive change in functionality … as if they were running on the source computing system; [0043]; examiner further notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity or merely a virtual entity such that container allocation and/or application assignment may constitute a resource ... first resource allocation on first machine may be any allocation of memory (e.g., physical hardware) used by the application on the first machine 100 examiner notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity or merely a virtual entity such that container allocation and/or application assignment may constitute a resource ... first resource allocation on first machine may be any allocation of memory (e.g., physical hardware 105) used by the application on the first machine 100 to the second machine, e.g., FIG. 4, 405-1, 405-2).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 9-12, 15-18 and 26-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over WO 2013/002777 to Hsu in view of US Patent Publication 2010/0205602 to Zedlewski et al.

Regarding dependent claims 9 and 15, Hsu may fail to explicitly teach each and every feature of:

the VIOS partition (cl. 9)/computer program readable code (cl. 15) is operable to automatically override the resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines.

Zedlewski et al. teach:

the VIOS partition (cl. 9)/**computer program readable code** (cl. 15) **is operable to automatically override the second resource allocation indicated by the selected mobility setting in response to identifying unavailable resources on either the first or second machines** (Zedlewski et al., [0009]-[0011], [0018] … at any time, the operating system can preempt a thread and force it to give up the CPU on which it is running in order to run another thread, [0021], [0031]-[0034] … *rescheduling* may be triggered according to rules programmed into the schedule, according to user-input parameters or both, or disabled altogether ... one measure of anti-cooperative execution behavior will be violation of user-specified thread performance requirement; FIGS. 5-6 … alternative scheduling option, [0053] … scheduler 610 may deal with the situations according to one or more options … conditions that trigger intervention may be user-selected or pre-set; …. Examiner notes claim language does not restrict basis of/interpretation of "unavailable resources" such that it the resource may be unavailable for any reason, i.e., error or physically unavailable).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). As both Hsu and Zedlewski pertain to efficient and effective data management and transfer mechanisms, it would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claims 10 and 16, Hsu may fail to explicitly teach each and every feature of:

an allocation module (cl. 10)/computer program readable code (cl. 16) operable to identify a memory resource allocation for the migration based on the selected mobility setting.

Zedlewski et al. teach:

an allocation module (cl. 10)/**computer program readable code** (cl. 16) **operable to identify a memory resource allocation for the migration based on the selected mobility setting** (Zedlewski et al., ., [0009]-[0011] … concurrent execution …; [0018], [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). As both Hsu and Zedlewski pertain to efficient and effective data management and transfer mechanisms, it would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claims 11 and 17, Hsu may fail to explicitly teach each and every feature of:

the allocation module (cl. 11)/computer program readable code (cl. 17) is operable to identify a processor resource allocation for the migration based on the selected mobility setting.

Zedlewski et al. teach:

the allocation module (cl. 11)/computer program readable code (cl. 17) **is operable to identify a processor resource allocation for the migration based on the selected mobility setting** (Zedlewski et al., [0009]-[0011] … concurrent execution … ; [0018], [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state; examiner notes claim language fails to provide any restrictive/condition regarding interpretation “processor resource allocation” or “selected mobility setting”).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks

(e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claims 12 and 18, Hsu may fail to explicitly teach each and every feature of:

the interface (cl. 12)/computer program readable code (cl. 18) is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation.

Zedlewski et al. teach:

the interface (cl. 12)/**computer program readable code** (cl. 18) **is configured to indicate a first mobility setting corresponding to a performance-based resource allocation and a second mobility setting corresponding to a concurrency-based resource allocation** (Zedlewski et al., [0009]-[0011] … concurrent execution …; [0018], [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … various settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state; [0096] e.g., users can select from choices for each VM ... any sharing, no sharing or internal sharing only .. default setting; Examiner notes that claim language does not specify any definition/restriction/conditions on the interpretation of the first and the second "mobility setting" as any setting can be interpreted as being performance based as the claim provides no measure regarding “performance based” … good performance, bad performance, etc., and similarly the claim language fails to set forth any definition for “concurrency-based” resource allocation and thus any allocation may be concurrency based as the claim language fails to provide any basis/condition for concurrency based, e.g., when nothing else is concurrently running/being processed, when x threads are being processed, etc.).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding independent claim 26, Hsu teaches:

A system, comprising: a first machine having a plurality of logical partitions (LPARs) each running an instance of an operating system and an application (Hsu, FIGS. 1, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface* … provides flexibility for users to selectively migrate running applications to external operating system environments in order to perform maintenance on the host operating system environment);

a management console operable to manage migrating the plurality of LPARs from the first machine to a second machine, the management console further operable to (Hsu, FIG. 5 … process management subsystem 535, process containerization module 530, live migration orchestrator 535):

display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Hsu, FIGS. 1, 3, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0032] assignment of

applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface*); [0032], [0043] … examiner notes claim language does not define “desired” such that any setting may be deemed “desired”; examiner notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity or merely a virtual entity such that container allocation and/or application assignment may constitute a resource);

receive, via the interface, a selection of a mobility setting to apply for migrating the LPARs (Hsu, FIGS. 1, 3, 4; [0030] seamless migration of the running applications (105-1 to 105-3) to a new computing system; [0032] assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems; [0043] operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface*; [0032], [0043])**; and**

initiate migration of the LPARs from the first machine to the second machine utilizing resources as set by the selected mobility setting (Hsu, FIGS. 1-4; [0033] once applications have been assigned to respective containers, the operating system may transfer the applications to their respective containers; [0040] from perspective of a user or client of the applications migrated from the first computing system, it will appear as if the applications have been running continuously without any substantive change in functionality … as if they were running on the source computing system; [0043]; examiner further notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity or merely a virtual entity such that container allocation and/or application assignment may constitute a resource ... first resource allocation on first machine may be any allocation of memory (e.g., physical hardware) used by the application on the first machine 100 examiner notes claim language does not specify what qualifies or does not qualify as a "resource" or even if such resource is necessary a physical entity

or merely a virtual entity such that container allocation and/or application assignment may constitute a resource ... first resource allocation on first machine may be any allocation of memory (e.g., physical hardware 105) used by the application on the first machine 100 to the second machine, e.g., FIG. 4, 405-1, 405-2 ...).

While Hsu teaches while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

wherein a first mobility setting sets a first resource allocation to accommodate a desired rate of migration, and wherein a second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations.

Zedlewski et al. teach:

wherein a first mobility setting sets a first resource allocation to accommodate a desired rate of migration, and wherein a second mobility setting sets a second resource allocation to accommodate a desired concurrency of LPAR migrations (Zedlewski et al., [0009]-[0011] … concurrent execution …; [0018], [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … various settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state; [0096] e.g., users can select from choices for each VM ... any sharing, no sharing or internal sharing only .. default setting; Examiner notes that claim language does not specify any definition/restriction/conditions on the interpretation of the “concurrency-based” resource allocation and thus any allocation may be concurrency based as the claim language fails to provide any basis/condition for concurrency based, e.g., when nothing else is concurrently running/being processed, when x threads are being processed, etc.).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate

applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 27, while Hsu teaches that assignment of applications determined by operating system based on heuristics and/or rules ... may determine the amount of processing resources utilized by each of the running applications, determine the amount of available processing resources in external computing systems and compartmentalize the applications into the containers by matching applications and containers to the external computing systems and that the operating system containers allow a user to *create the operating system containers and assign running applications to the created containers based on input to the host computing system through a user interface* (Hsu, [0032], [0043]), Hsu et al. may fail to explicitly teach each and every feature of:

each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration.

Zedlewski et al. teach:

each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration (Zedlewski et al., [0009]-[0010] … standing goal of all computer design is to enable applications to run as fast and as efficiently as possible, even when sharing system resources and one way to accomplish this is efficient schedule for executing threads; [0018], [0074], [0094]-[0099] allocate

suitable resources … threads count exceeds).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 28, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

an allocation module configured to, in response to detecting a mismatch between the amount of memory on the first machine and the amount of memory on the second machine to allocate to the migration based on the selected mobility setting, negotiate a balance of resource allocation on the first and second machines based on available resources on the first and second machines.

Zedlewski et al. teach:

an allocation module configured to, in response to detecting a mismatch between the amount of memory on the first machine and the amount of memory on the second machine to allocate to the migration based on the selected mobility setting, negotiate a balance of resource allocation on the first and second machines based on available resources on the first and second machines ([0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine

monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 29, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

the management console is operable to receive a different selected mobility setting for each of the plurality of LPARs and automatically apply the respective selected mobility settings to the respective migrations of the LPARs.

Zedlewski et al. teach:

the management console is operable to receive a different selected mobility setting for each of the plurality of LPARs and automatically apply the

respective selected mobility settings to the respective migrations of the LPARs (Zedlewski et al., [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0098] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 30, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

each selectable mobility setting sets a run time for the threads.

Zedlewski et al. teach:

each selectable mobility setting sets a run time for the threads (Zedlewski et al., [0021] … known schedulers may adjust the amount of execution time allocated to each of a set of runnable threads. … Examiner further notes, however, that the claim language does not distinguish features of run time such that all the run times may be the same and may even be a default max time after which, e.g., an error/failure may be identified).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 31, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

an allocation module configured to evaluate available resources on the first and second machines based on the selected mobility setting.

Zedlewski et al. teach:

an allocation module configured to evaluate available resources on the first and second machines based on the selected mobility setting (Zedlewski et al., [0018] … the OS designates which software threads the logical processors are to

execute and can also issue commands to cause an idle logical processor to be put in a halt state such that execution resources are made available for use by any remaining logical processors (examiner notes such a determination to cause an idle logical processor to be put in a halt state inherently requires an 'evaluation' of available resources); [0031] … rescheduling … upon sensing the rescheduling condition … ; … Examiner further notes, however, that the claim language does not set forth features of ‘evaluate’ such that any processing may be deemed to be evaluating available resources between the identified machines).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Regarding dependent claim 32, while Hsu teaches determining an amount of available resources and balancing workloads (Hsu [0032], [0044]), Hsu may fail to teach each and every feature of:

each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration; and further comprising an allocation module configured to determine availability of the amount of memory and the quantity of threads on the first and second machines for the migration based on the selected mobility setting and in response to determining an unavailability of the amount of

memory or the quantity of threads on either the first or second machines for the migration, negotiate a balance of memory and threads to use on the first and second machines to use for the migration.

Zedlewski et al. teach:

each selectable mobility setting sets an amount of memory on the first and second machines to allocate to the migration and a quantity of threads to use on the first and second machines for the migration; and further comprising an allocation module configured to determine availability of the amount of memory and the quantity of threads on the first and second machines for the migration based on the selected mobility setting and (Zedlewski et al., [0072] memory allocated to that VM, [0073] some interface (e.g., virtual machine monitor VMM) is usually required between a VM and host platform, which is responsible for executing instructions and transferring data to and from the memory and storage devices, [0074] … VMM may be set up to expose “generic" devices and facilitate VM migration and hardware platform independence; [0094]-[0099] user choices … settings ... user may specify not only a percentage of a package's time, but also indicate to the scheduler 610, via the console 300 or otherwise, such as with settings in a file or other user-specified state; thread count exceeds).

in response to determining an unavailability of the amount of memory or the quantity of threads on either the first or second machines for the migration, negotiate a balance of memory and threads to use on the first and second machines to use for the migration (Zedlewski et al., [0018] … the OS designates which software threads the logical processors are to execute and can also issue commands to cause an idle logical processor to be put in a halt state such that execution resources are made available for use by any remaining logical processors (examiner notes such a determination to cause an idle logical processor to be put in a halt state inherently requires an 'evaluation' of available resources); [0031] … rescheduling … upon sensing the rescheduling condition … ; … Examiner further notes, however, that the claim language does not set forth features of ‘negotiate’ such that any allocation/processing may be deemed to be negotiation available resources among the machines).

Hsu pertains to methods and systems of migrating applications with dynamic operating system containers in which a number of applications currently executed within a first operating system environment are identified for migration and migrated to a second operating system environment, thereby allowing users to selectively migrate applications to other operating system environments in order to perform maintenance

on the host operating system or balance workloads among multiple computing systems (Hsu, Abstract, [0043]). Zedlewski et al. pertains to mechanisms for dealing with shared resource hoarding in multi-threaded architectures that enable users to control as least some decision about what to do in anti-cooperative process in a multi-threaded architecture (Zedlewski et al., [0029]). It would have been obvious for one of ordinary skill in the art at the time of applicant's invention to incorporate the intervening resource allocation approaches (e.g., user input) for rescheduling/reallocating resources as set forth in Zedlewski et al. with the migration system/methods of Hsu in order to reduce run-time and increase use efficiency by effectively using resources to complete tasks (e.g., migration) by not wasting a pre-allocated execution time of a given thread, but rather than, e.g., keeping resources idle based on previously set settings/allocations (Zedlewski et al., [0021], [0031]-[0033], [0053], [0054]).

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 7, 8, 13 and 14 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 2012/0216135 to Wong et al.

Regarding claims 7 and 13, Wong et al. teach:

A system (Wong et al., FIGS. 1A-1B; system 100)**, comprising:**

a first machine having a running application (Wong et al., FIG. 1A, 1B, one of the host computers 104; [0017]-[0018] … host computers 104 … virtual machines running within each host computer 104);

a management console operable to manage migrating the running application from the first machine to a second machine, the management console

further operable to display an interface comprising a plurality of selectable mobility settings, each mobility setting corresponding to a desired resource allocation to utilize for the migration (Wong et al., FIGS. 1A-3, 7A, 7B; [0018]-[0019] ...virtual machine management center 102; … VM management center 102 manages the virtual infrastructures, including managing the host computers 104, the virtual machines running within each host computer 104, provisioning, migration, resource allocations; [0021] virtual machine monitor 125, which is a software interface layer that enables sharing of the hardware resources of host computer 104 by the virtual machines; [0026] ... allows disk images of virtual machines to be migrated between storage arrays without interrupting the virtual machine whose disk image is being migrated or any applications running inside it; [0032] GUI, command line interface; [0042]-[0043] method 510 at step 502 … graphical user interface includes receiving a mouse click on one or more objects in the graphical user interface, i.e., to select one the one or more objects. … the VI client can include a graphical user interface that allows a user manually migrate a disk image of a virtual machine between storage arrays; [0045] “Get VM1” … “Move”);

receive, via the interface, a selection of a mobility setting to apply for migrating the running application (Wong et al., FIGS. 1A-3, 7A, 7B; [0018]-[0019] ...virtual machine management center 102; … VM management center 102 manages the virtual infrastructures, including managing the host computers 104, the virtual machines running within each host computer 104, provisioning, migration, resource allocations; [0021] virtual machine monitor 125, which is a software interface layer that enables sharing of the hardware resources of host computer 104 by the virtual machines; [0026] ... allows disk images of virtual machines to be migrated between storage arrays without interrupting the virtual machine whose disk image is being migrated or any applications running inside it; [0032] GUI, command line interface; [0042]-[0045] method 510 at step 502 … graphical user interface includes receiving a mouse click on one or more objects in the graphical user interface, i.e., to select one the one or more objects … the VI client can include a graphical user interface that allows a user manually migrate a disk image of a virtual machine between storage arrays); **and**

initiate migration of the running application from the first machine to the second machine utilizing resources as set by the selected mobility setting (Wong et al., FIGS. 1A-3, 7A, 7B; [0018]-[0019] ...virtual machine management center 102; … VM management center 102 manages the virtual infrastructures, including managing the host computers 104, the virtual machines running within each host computer 104, provisioning, migration, resource allocations; [0026] ... allows disk images of virtual machines to be migrated between storage arrays without interrupting the virtual machine whose disk image is being migrated or any applications running inside it;

[0032] GUI, command line interface; [0042]-[0045] to accomplish the task of migrating the first VM the first host computer to the second host computer, the user may click on an object representing the first VM and drag the object to another portion of the visual hierarchy. In some embodiments, the user interface can display only technically feasible or recommended destinations for such a VM migration. … the VI client can include a graphical user interface that allows a user to manually migrate disk images of a virtual machine between storage arrays, as described in FIG. 1B).

Regarding claims 8 and 14, Wong et al. teach:

at least one virtual input/output server (VIOS) partition operable to negotiate a balance of resource allocations between the first machine and the second machine based on the selected mobility setting (Wong et al., [0017] virtualization software partitions a physical host computer 104 into multiple secure and portable virtual machines that run on the same physical server).

Conclusion

Applicant's amendment necessitated any new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Examiner has cited particular columns and line numbers (or paragraphs) in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific imitations within the individual claim, other passages and figures may apply as well. More particularly, e.g., in the instances the Examiner has identified Figures of the applied prior art reference, it is understood that the corresponding portion of the written description describing the identified Figures is relied upon. It is respectfully requested from the Applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. **The entire reference(s) is/are to be considered to provide disclosure relating to the claimed invention.**

Any inquiry concerning this communication or earlier communications from the examiner should be directed to MARYAM IPAKCHI whose telephone number is (571)270-3237. The examiner can normally be reached on M-(alt)F 9-3:00EST. **Any interview requests should be made via an Interview Agenda faxed to the examiner at (571)270-4237. The Interview Agenda should include claims/items to be discussed and proposed dates/times,**

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matt M. Kim can be reached on (571)272-4182. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/MARYAM IPAKCHI/
Examiner
Art Unit 2171

/MATT KIM/
Supervisory Patent Examiner, Art Unit 2171

Notice of References Cited	Application/Control No. 13/671,422	Applicant(s)/Patent Under Reexamination GARZA ET AL.	
	Examiner MARYAM IPAKCHI	Art Unit 2171	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-6,105,053 A	08-2000	Kimmel et al.	718/105
*	B	US-2005/0060722 A1	03-2005	Rochette et al.	719/319
*	C	US-2006/0259606 A1	11-2006	Rogers et al.	709/223
*	D	US-2007/0250838 A1	10-2007	Belady et al.	718/105
*	E	US-2009/0132804 A1	05-2009	Paul et al.	713/150
*	F	US-7,814,491 B1	10-2010	Chen et al.	718/104
*	G	US-2010/0318608 A1	12-2010	Huang et al.	709/205
*	H	US-8,875,160 B2	10-2014	Hunt et al.	719/319
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Chahal, et al. "Testing Live Migration with Intel Virtualization Technology FlexMigration", retrieved at: https://communities.intel.com/docs/DOC-2538, White Paper, Jan. 2009, pages 1-12
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.